

UNIVERSIDAD POLITÉCNICA SALESIANA

FACULTAD DE INGENIERÍAS

SEDE QUITO- CAMPUS SUR

CARRERA DE INGENIERÍA DE SISTEMAS

MENCIÓN TELEMÁTICA

DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE GEOREFERENCIACION DE LA CIUDAD DE QUITO EMPLEANDO JSF Y CONEXIÓN CON GOOGLE EARTH

TESIS PREVIA LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS

ANILOA GUERRERO JOHANNA PAULINA

JUMBO BALCAZAR EDISON RODRIGO

DIRECTOR. ING. GUSTAVO NAVAS

Quito, Julio 2010

DECLARACIÓN

Nosotros, Aniloa Guerrero Johanna Paulina, Jumbo Balcázar Edison Rodrigo declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la universidad politécnica salesiana, según lo establecido por la ley de propiedad intelectual, por su reglamento y por la normatividad institucional vigente.

Aniloa Guerrero Johanna Paulina.

Jumbo Balcázar Edison Rodrigo

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Aniloa Guerrero Johanna Paulina y Jumbo Balcázar Edison Rodrigo, bajo mi dirección.

Ing. Gustavo Navas R.

Director de tesis

AGRADECIMIENTOS

Esta tesis, si bien ha requerido de esfuerzo y mucha dedicación por parte de sus autores y su director de tesis, no hubiese sido posible su finalización sin la cooperación desinteresada de todas y cada una de las personas que me han acompañado en la culminación de esta etapa, las cuales han sido un soporte muy fuerte en momentos de angustia y desesperación.

Por ello, es para mí un verdadero placer utilizar este espacio para ser justa y consecuente con ellas, expresándoles mis agradecimientos.

A **Dios**, por estar a mi lado, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y ayuda.

Un agradecimiento a una persona muy especial Suor Chiara por estar siempre a mi lado, ser mi respaldo, soporte y consejera durante todas las etapas de mi vida

Debo agradecer de manera sincera al Ing. Gustavo Navas por haber aceptado la dirección de esta tesis, colaborar con su esfuerzo, confianza en nosotros y facilitar siempre los medios suficientes para llevar a cabo todas las actividades propuestas durante su desarrollo.

A mi compañero de tesis Sr Edison Jumbo con quien logramos desarrollar de manera exitosa este proyecto, venciendo todas y cada una de las adversidades que se nos presentaron.

Una agradecimiento especial a mi mis padres y a mi familia por su apoyo, colaboración e inspiración ya que sin ellos no habría sido posible llevar a cabo este proyecto.

En general quisiera agradecer a todos y cada uno de los ingenieros y amigos, que no necesito nombrar porque tanto ellos como yo sabemos que desde los más profundo de su corazón me han brindado todo el apoyo, colaboración, ánimo, enseñanza pero sobre todo cariño y amistad

DEDICATORIA

Esta tesis va dedicada a todas y cada una de esas personas que siempre estuvieron ahí presentes en cada uno de esos momentos especiales.

Todo este esfuerzo que he realizado durante mi carrera y tesis, se las dedico con todo mi corazón para que ustedes entiendan que sin su apoyo no lo hubiera logrado.

De manera especial va dedicado en particular a mis padres José Alfonso Anilao Toaquiza, Janeth de los Dolores Guerrero Espinoza y Suor Chiara quienes han sido la razón de mi inspiración, esfuerzo y constancia a lo largo de mi vida. Gracias de corazón por su apoyo tan incondicional y sincero

DEDICATORIA

Este trabajo está dedicado a las personas más importantes en mi vida, aquellas que siempre han estado presente alentándome todos los días, quienes han sabido apoyarme y guiarme, dándome fuerzas para seguir adelante, a quienes más respeto, a los cuales les debo todo, por quienes tengo ganas de seguir hasta el final.

A mis padres.

Y en especial a las personas que más adoro en este mundo, por quien todo se vuelve posible a ti mi amor Cristina Villavicencio.

RESUMEN

CAPÍTULO I. “Referencia Teórica”.- Este capítulo contiene una descripción de los temas que se consideran importantes para el inicio del sistema, así como las referencias y funcionamientos de las diferentes tecnologías empleadas en el desarrollo del sistema, para entender cuál será la función del proyecto.

CAPÍTULO II. “Determinación de Requerimiento y Análisis”.- Describe cuales son los requerimientos del sistema y además provee de información necesaria para el desarrollo del mismo mostrando cuales serán los autores y los subsistemas con los que contara el proyecto, así como los permisos que tiene cada usuario.

CAPÍTULO III. “Diseño del Sistema”.- Muestra como está estructurado el proyecto, el numero de clases que posee, las interfaces de usuario y los debidos procesos que tiene que seguir para un mejor resultado y los mapas de navegación que tiene cada usuario.

CAPÍTULO IV. “Construcción”.- Indica cuales son los principales pasos que debe seguir el programador para construir una aplicación web con tecnología JSF y pautas que debe de seguir para su correcto funcionamiento.

CAPÍTULO V. “Conclusiones y Recomendaciones”.- Este capítulo expone cuales son las conclusiones y recomendaciones a las que se han llagado después del análisis, diseño y creación del sistema.

ÍNDICE

| | |
|---|------|
| DECLARACIÓN | II |
| CERTIFICACIÓN | III |
| AGRADECIMIENTOS | IV |
| DEDICATORIA..... | VI |
| DEDICATORIA..... | VII |
| RESUMEN | VIII |
| ÍNDICE | IX |
| ÍNDICE DE GRÁFICOS | XIII |
| ÍNDICE DE TABLAS | XV |
| CAPÍTULO I | 1 |
| REFERENCIA TEÓRICA | 1 |
| 1.1. QUITO..... | 1 |
| 1.1.1. CARACTERÍSTICAS..... | 1 |
| 1.2. GOOGLE EARTH..... | 2 |
| 1.2.1. BENEFICIOS DE GOOGLE EARTH | 3 |
| 1.2.2. VENTAJAS..... | 3 |
| 1.3. GOOGLE MAPS | 3 |
| 1.4. XML (EXTENSIBLE MARKUP LANGUAGE) | 4 |
| 1.4.1. HISTORIA DE XML | 4 |
| 1.4.2. VENTAJAS DE XML..... | 5 |
| 1.5. KML..... | 5 |
| 1.6. LENGUAJE DE PROGRAMACION JAVA..... | 8 |
| 1.6.1. ¿QUE ES JAVA?..... | 8 |

| | |
|---|----|
| 1.6.2. HISTORIA..... | 8 |
| 1.6.3. MÁQUINA VIRTUAL JAVA (JAVA VIRTUAL MACHINE, JVM)..... | 9 |
| 1.6.4. CARACTERÍSTICAS DE JAVA | 10 |
| 1.7. JSP..... | 11 |
| 1.7.1. MOTOR JSP | 12 |
| 1.8. JSF | 13 |
| 1.8.1. FUNCIONAMIENTO JSF | 13 |
| CAPITULO II | 15 |
| DETERMINACIÓN DE REQUERIMIENTOS Y ANÁLISIS | 15 |
| 1.1. REQUERIMIENTOS BÁSICOS..... | 15 |
| 1.2. REQUERIMIENTOS FUNCIONALES | 16 |
| 1.3. REQUERIMIENTOS NO FUNCIONALES..... | 17 |
| 1.3.1. REQUERIMIENTOS DEL SOFTWARE..... | 17 |
| 1.4. COMPRENSIÓN DEL PROCESO | 18 |
| 1.5. IDENTIFICACIÓN DE LOS DATOS EMPLEADOS E INFORMACIÓN GENERADA..... | 18 |
| 1.6. FRECUENCIA Y VOLUMEN DEL PROCESO | 19 |
| 1.7. REQUERIMIENTOS DE LAS TRANSACCIONES DE LOS USUARIOS | 19 |
| 1.8. REQUERIMIENTOS DE DECISIÓN DE LOS USUARIOS..... | 20 |
| 1.9. ANÁLISIS DEL SISTEMA | 21 |
| 1.9.1. ORGANIZACIÓN | 21 |
| 1.9.2. SISTEMA..... | 21 |
| 1.9.3. PARTICIPANTES | 22 |
| 1.9.4. ACTORES. | 23 |
| 1.9.5. CASOS DE USO | 24 |
| CAPÍTULO III | 31 |
| DISEÑO DEL SISTEMA..... | 31 |

| | |
|--|----|
| 3.1. CONCEPTOS Y PRINCIPIOS | 31 |
| 3.2. EL DISEÑO DE LOS DATOS..... | 31 |
| 3.3. CLASES | 32 |
| 3.3.1. CLASES | 33 |
| 3.4. PRINCIPALES TAREAS DEL SISTEMA..... | 45 |
| 3.4.1. REGISTRO DE USUARIOS. | 46 |
| 3.4.2. AUTENTICACIÓN DE USUARIOS..... | 47 |
| 3.4.3. INGRESO DE INFORMACIÓN..... | 49 |
| 3.4.4. TRATAMIENTO DE LA INFORMACIÓN | 51 |
| 3.4.5. ASIGNACIÓN DE INFORMACIÓN..... | 53 |
| 3.4.6. BÚSQUEDA DE INFORMACIÓN | 55 |
| 3.5. EL DISEÑO ARQUITECTÓNICO..... | 59 |
| 3.6. DISEÑO DEL SISTEMA..... | 59 |
| 3.7. EL DISEÑO DE LA INTERFAZ. | 60 |
| 3.7.1. INTERFAZ USUARIO VISITANTE | 60 |
| 3.7.2. INTERFAZ DE USUARIOS REGISTRADOS..... | 61 |
| 3.7.3. INTERFAZ USUARIO ADMINISTRADOR..... | 62 |
| 3.8. DISEÑO DE LA SALIDA. | 62 |
| 3.9. DISEÑO DE LA BASE DE DATOS | 63 |
| 3.9.1. USUARIO VISITANTE..... | 65 |
| 3.9.2. USUARIO REGISTRADO..... | 66 |
| 3.9.3. USUARIO ADMINISTRADOR | 67 |
| CAPÍTULO IV | 68 |
| CONSTRUCCIÓN | 68 |
| 4.1. ENTORNO DE DESARROLLO INTEGRADO..... | 68 |
| 4.2. VISUAL WEB JSF | 69 |
| 4.3. FRAGMENTOS..... | 70 |

| | |
|---|----|
| 4.4. SESIONES..... | 73 |
| 4.5. JDOM (CREACIÓN DE ARCHIVOS KML)..... | 75 |
| 4.6. ARCHIVOS WAR | 77 |
| 4.7. CONEXIÓN BASE DE DATOS | 78 |
| CAPITULO V | 80 |
| CONCLUSIONES Y RECOMENDACIONES | 80 |
| 5.1. CONCLUSIONES..... | 80 |
| 5.2. RECOMENDACIONES | 82 |
| BIBLIOGRAFÍA | 83 |
| ANEXOS | 84 |

ÍNDICE DE GRÁFICOS

| | |
|---|----|
| Figura 1. Ejemplo KML..... | 7 |
| Figura 2. Logotipo del lenguaje de programación Java..... | 8 |
| Figura 3. Funcionamiento del JSP | 12 |
| Figura 4. Caso de Uso Gestión de Información..... | 26 |
| Figura 5. Caso de Uso Gestión de Usuarios | 28 |
| Figura 6. Caso de Uso Gestión de sugerencias y comentarios..... | 30 |
| Figura 7. Diagrama de clases autor visitante | 32 |
| Figura 8. Clases usuario registrado..... | 35 |
| Figura 9. Administración datos del usuario registrado..... | 36 |
| Figura 10. Búsqueda del usuario registrado..... | 38 |
| Figura 11. Clases usuario administrador y administración de usuarios..... | 41 |
| Figura 12. Ingreso de datos Fuente: El autor | 42 |
| Figura 13. Clases de uso Comun y Búsqueda de datos | 44 |
| Figura 14. Diagrama de actividad registro de Usuarios..... | 46 |
| Figura 15. Diagrama de secuencia registro de Usuarios..... | 47 |
| Figura 16. Diagrama de actividad de Autenticación de usuarios..... | 48 |
| Figura 17. Diagrama de secuencia de Autenticación de usuarios..... | 49 |
| Figura 18. Diagrama de actividad ingreso de información | 50 |
| Figura 19. Diagrama de secuencia ingreso de información | 50 |
| Figura 20. Diagrama de actividad actualización de información..... | 51 |
| Figura 21. Diagrama de secuencia actualización de información..... | 52 |
| Figura 22. Diagrama de actividad eliminación de información | 52 |
| Figura 23. Diagrama de secuencia eliminación de información | 53 |
| Figura 24. Diagrama de actividad asignación de informació | 54 |
| Figura 25. Diagrama de actividad asignación de información | 54 |
| Figura 26. Diagrama de actividad búsqueda sin beneficios | 55 |
| Figura 27. Diagrama de secuencia búsqueda sin beneficios | 56 |

| | |
|--|----|
| Figura 28. Diagrama de actividad búsqueda con beneficios | 57 |
| Figura 29. Diagrama de secuencia búsqueda con beneficios | 58 |
| Figura 30. Funcionamiento interno del sistema | 59 |
| Figura 31. Interfaz de usuario visitante | 59 |
| Figura 32. Interfaz de usuario visitante | 60 |
| Figura 33. Interfaz de usuario registrado..... | 61 |
| Figura 34. Interfaz de usuario Administrador..... | 62 |
| Figura 35. Modelo conceptual de la base de datos | 63 |
| Figura 36. Modelo físico de la base de datos | 64 |
| Figura 37. Mapa de sitio usuario visitante | 65 |
| Figura 38. Mapa de sitio usuario registrado | 66 |
| Figura 39. Mapa de sitio usuario registrado | 67 |
| Figura 40. Interfaz Grafica de Netbeans | 69 |
| Figura 41. Visual Web JSF | 70 |
| Figura 42. Add Binding Attribute..... | 70 |
| Figura 43. Page Fragment Box | 71 |
| Figura 44. Selección de Fragmentos..... | 71 |
| Figura 46. Selección de Fragmentos..... | 72 |
| Figura 45. Fragmento nuevo | 72 |
| Figura 47. Relaciones entre interfaces..... | 73 |
| Figura 48. Relaciones entre interfaces..... | 74 |
| Figura 49. Creación de sesiones..... | 74 |
| Figura 50. Búsqueda de usuarios registrados..... | 74 |
| Figura 51. Búsqueda de usuarios registrados..... | 75 |
| Figura 52. Archivo KML..... | 76 |
| Figura 53. Diseño del KML..... | 76 |
| Figura 54. Creación del archivo KML | 77 |
| Figura 55. Creación del archivo War | 77 |
| Figura 56. Datos conexión a la base de datos | 78 |
| Figura 57. Datos conexión a la base de datos | 78 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1. Organización | 21 |
| Tabla 2. Sistema | 21 |
| Tabla 3. Participantes en el desarrollo del sistema | 22 |
| Tabla 4. Autor Visitante | 23 |
| Tabla 5. Autor Registrado | 23 |
| Tabla 6. Autor Administrador..... | 24 |
| Tabla 7. Caso de Uso Gestión de Información..... | 26 |
| Tabla 8. Caso de Uso Gestión de usuarios..... | 27 |
| Tabla 9. Caso de Uso Gestión de sugerencias y comentarios..... | 29 |
| Tabla 10. Clases Fragmentos usadas en el sistema..... | 34 |
| Tabla 11. Clases de uso del usuario visitante | 34 |
| Tabla 12. Clases de uso del usuario registrado | 37 |
| Tabla 13. Clases de uso del usuario administrador..... | 40 |

CAPÍTULO I

REFERENCIA TEÓRICA

1.1. QUITO

1.1.1. CARACTERÍSTICAS

En 1534 se fundó la ciudad Quito. Con una arquitectura estilo colonial por lo cual en 1978 fue declarada por la UNESCO Patrimonio Cultural de Humanidad

En la década de 1930, las clases altas del centro de la ciudad se desplazaron al norte. Surgieron barrios residenciales dentro del esquema de "ciudad jardín". Los espacios del centro fueron ocupados por inmigrantes de las provincias vecinas. La parte antigua de la ciudad pudo por consiguiente conservar su traza original y su arquitectura colonial enriquecida con los nuevos aportes de los siglos XIX y XX. Hacia la mitad del siglo XX, el espacio urbano estaba ya socialmente estratificado.

La extensión de la ciudad hacia el norte y el sur comenzó durante los años 1980, cuando la principal área turística en la parte central norte comenzó a crecer. Quito, es hoy una metrópoli emprendedora y el centro político del país.

La ciudad no contó con un plan para su crecimiento, por lo cual muchas de las partes de la ciudad, como los barrios a los extremos tienen inconvenientes para su acceso y localización.

La ciudad actualmente se ha convertido en una metrópolis en la que la mayoría de personas forasteras a este, en la mayoría de las veces pierden el rumbo o no dan con las direcciones indicadas por lo cual radica la importancia del sistema.

1.2. GOOGLE EARTH

Actualmente es el buscador más usado en el mundo por su rapidez, sencillez y efectividad en las búsquedas. En la tesis es fundamental la utilización de Google Earth ya que permite navegar virtualmente a cualquier parte de la ciudad Quito, esta herramienta incorpora fotos tomadas desde satélite obteniendo así información de avenidas, restaurantes, hoteles, parques en lugares en los cuales será enfocada la tesis para la búsqueda.

Para el desarrollo del proyecto hemos utilizado la versión básica de Google Earth (Google Earth Free) la cual también nos permite salvar las búsquedas y anotaciones y compartirlas en formato **KML**.

En la investigación se ha encontrado tres tipos de versiones las mismas que tienen un pago adicional las cuales son:

- Google Earth Plus Es una ampliación del programa básico pensado para uso particular cuesta 20 dólares al año y ofrece una mayor resolución de las imágenes satelitales. Google Earth Plus viene a ser una versión útil para quienes hacen un uso intensivo de los mapas y del GPS.
- Google Earth Pro Versión profesional con precio de 400 dólares diseñada específicamente para empresas. Incorpora muchas mejoras como por ejemplo control del tráfico o importación desde hojas de cálculo. Permite el acceso a terabytes de información que son descargados según las peticiones del usuario. Mejora aún más la resolución de las imágenes, incluye soporte telefónico, importación de sistemas de información geográfica.
- Google Earth Enterprise es una tecnología para integrarlo con el sistema de información geográfica de la empresa. “Planteable para compañías que se dediquen a este tipo de software.”¹

¹ http://www.error500.net/google_earth_pro_plus_enterprise?page=1

1.2.1. BENEFICIOS DE GOOGLE EARTH

Rápido: Google Earth utiliza una sofisticada tecnología de reproducción de los datos geográficos que dispone y son de gran riqueza para los usuarios a una velocidad de respuesta adecuada.

Completo: el sistema Google Earth es capaz de alojar bases de datos masivas con diversos tipos de datos geográficos. El usuario puede crear un planeta 3D o un mapa 2D para todos los datos geoespaciales, además de fichas de búsqueda personalizadas para explorar los atributos geográficos o las bases de datos.

Flexible: Google Earth funciona en combinación con distintos datos y sistemas GIS tradicionales. Google Earth se puede utilizar para publicar datos almacenados en estos sistemas en toda la organización

1.2.2. VENTAJAS

Con google earth es posible atender a cientos de usuarios diarios desde un único servidor y a miles desde un pequeño clúster.

Posee funciones de búsqueda que permiten a los usuarios buscar atributos de datos geográficos y bases de datos.

Las diferentes búsquedas es posible hacerlo mediante herramientas integradas para enviar vistas, marcas de posición y anotaciones por correo electrónico gracias al lenguaje KML, un formato XML de gran flexibilidad

Una aclaración importante es que las imágenes satelitales tendrán mayor o menor resolución dependiendo de la importancia urbana que tenga la ciudad.

1.3. GOOGLE MAPS

Google Maps es un servicio que muestra mapas o croquis de ciudad, Esta herramienta fue diseñada como un complemento de google Local.

En el 2005 comenzó a ofrecer además imágenes satelitales que son las similares a las mostradas en google earth. Además nos permite crear aplicaciones web utilizando las imágenes y mapas que tiene este servicio.

1.4. XML (EXTENSIBLE MARKUP LANGUAGE)

1.4.1. HISTORIA DE XML

El XML es un lenguaje de marcas que permite a diferentes aplicaciones interactuar a través de la red.

“Este estándar permite que los diseñadores creen sus propias etiquetas, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones”²

Una de las ventajas de XML es que en el mundo del internet permite que se comuniquen muchos sistemas que existen

La tecnología de XML (lenguaje de marcas extensible) fue diseñada para transportar datos y almacenar datos

XML es una tecnología en realidad muy sencilla que tiene a su alrededor otras tecnologías que la complementan y la hacen mucho más grande y con posibilidades mucho mayores.

Esta tecnología no se usa solamente en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

Hay que recordar que XML es la herramienta más común para las transmisiones de datos entre todo tipo de aplicaciones, es considerado como un software y hardware independiente para el almacenamiento y descripción de la información.

² <http://www.masadelante.com/faqs/xml>

Es extensible: Es posible con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin ninguna complicación.

Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones

“La tecnología XML busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible”³

1.4.2. VENTAJAS DE XML

El XML se puede usar para infinidad de trabajos y aporta muchas ventajas en amplios escenarios.

Comunicación de datos. Si la información se transfiere en XML, cualquier aplicación podría escribir un documento de texto plano con los datos que estaba manejando en formato XML y otra aplicación recibir esta información y trabajar con ella.

Migración de datos. Si tenemos que mover los datos de una base de datos a otra sería muy sencillo si las dos trabajasen en formato XML.

Aplicaciones web. Hasta ahora cada navegador interpreta la información a su manera. Con XML tenemos una sola aplicación que maneja los datos y para cada navegador o soporte podremos tener una hoja de estilo para aplicarle y tener un formato de salida adecuado.

1.5. KML

Esta herramienta de KML será utilizada para la transmisión de datos con Google earth y para ello es necesario conocer sus aplicaciones para las implementaciones a nuestro proyecto

³ <http://www.desarrolloweb.com/articulos/449.php>

KML es un lenguaje basado en marcas para representar datos geográficos en forma tridimensional.

Un fichero KML especifica una característica (un lugar, una imagen) dentro de Google Earth. Contiene título, una descripción básica del lugar, sus coordenadas (latitud y longitud) y alguna otra información.

Los archivos KML se pueden crear directamente en Google Earth, básicamente un archivo KML está estructurado de la siguiente forma:

- Un encabezado XML. Es la línea número uno de todos los archivos KML. Antes de esta línea no puede haber caracteres ni espacios.
- Una declaración de espacio de nombre de KML. Es la línea número dos de todos los archivos KML
- Un objeto de marca de posición (Placemark) que contiene los siguientes elementos:
 - Un nombre (*name*) que se utiliza como etiqueta para la marca de posición.
 - Una descripción (*description*) que aparece en una "viñeta" junto a la marca de posición.
 - Un punto (*Point*) que especifica la posición de la marca de posición en la superficie de la Tierra (*la longitud, la latitud y, opcionalmente, la altitud*).

EJEMPLO DE LA ESTRUCTURA DE UNA ARCHIVO KML

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.1">
<Document>
  <Placemark>
    <name>universidad</name>
    <address>Quito, Ecuador</address>
    <LookAt>
      <longitude>-78.55</longitude>
      <latitude>-0.283</latitude>
      <altitude>0</altitude>
      <range>500.050781</range>
      <tilt>0</tilt>
      <heading>0</heading>
    </LookAt>
    <Point>
      <coordinates>-78.52427700000001,-
0.229498,0</coordinates>
    </Point>
  </Placemark>
</Document>
</kml>
```

Figura 1. Ejemplo KML

Fuente: El Autor

1.6. LENGUAJE DE PROGRAMACION JAVA



Figura 2. Logotipo del lenguaje de programación Java

Fuente: http://www.cad.com.mx/historia_del_lenguaje_java.htm

1.6.1. ¿QUE ES JAVA?

Java es una plataforma virtual de software desarrollada por Sun Microsystems de tal manera que los programas creados en ella puedan ejecutarse sin cambios en diferentes tipos de arquitecturas y dispositivos computacionales.

1.6.2. HISTORIA

“Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principio de los años 90”⁴.

Java es un lenguaje con el cual se puede crear cualquier tipo de aplicación, no solo de escritorio, ya que su verdadero potencial esta en el internet por su plataforma J2EE ya que gracias a eso obtuvo su reconocimiento.

“El proyecto Green fue el primero en el que aplicó Java, y consistía en un sistema de control completo de los aparatos electrónicos y el entorno de un hogar. Con este fin se construyó un ordenador experimental denominado *7 (Star Seven). El sistema presentaba una interfaz basada en la representación de la casa de forma

⁴ http://www.cad.com.mx/historia_del_lenguaje_java.htm

animada y el control se llevaba a cabo mediante una pantalla sensible al tacto. En el sistema aparecía ya Duke, la actual mascota de Java.”⁵

El lenguaje java no fue presentado hasta en 1995, y su nombre original era OAK que significa roble en ingles pero ya era usado por otra casa comercial, cambiando su nombre al que ahora conocemos con una serie de modificaciones para adaptarlo al internet, un verdadero campo de competencia para la Sun Microsystems.

Java es uno de los lenguajes de programación que más aceptación ha tenido, a pesar de sus difíciles inicios muchos programadores se han orientado a este, ya que desde el principio se daba como un lenguaje multiplataforma.

La plataforma java consta de las siguientes partes:

- El lenguaje de programación.
- La máquina virtual de java.
- El api java, una biblioteca estándar para el lenguaje.

1.6.3. MÁQUINA VIRTUAL JAVA (JAVA VIRTUAL MACHINE, JVM)

Es un programa en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial, el cual es generado por el compilador del lenguaje.

Gracias a la maquina virtual se puede usar el mismo fichero .class creado en Windows en Gnu/Linux, sin cambios. Lo cual da una gran ventaja y demuestra su arquitectura multiplataforma.

“La Máquina Virtual Java (JVM) es el núcleo del lenguaje de programación Java. De hecho, es imposible ejecutar un programa Java sin ejecutar alguna implantación de la JVM. En la JVM se encuentra el motor que en realidad ejecuta

⁵ http://pisuerga.inf.ubu.es/Isi/Invest/Java/Tuto.Oct98/1_Intro/2_Histor.htm

el programa Java y es la clave de muchas de las características principales de Java, como la portabilidad, la eficiencia y la seguridad.”⁶

Cualquier aplicación java se puede ejecutar en cualquier plataforma siempre y cuando exista la JVM.

1.6.4. CARACTERÍSTICAS DE JAVA

- **Lenguaje simple.-** Java es un lenguaje simple de aprender ya que es bastante familiar a los lenguajes como C o C++, así que los programadores habituados en estos lenguajes no tendrán mayores inconvenientes en el empleo de Java.

“Los programadores experimentados en C++ pueden migrar muy rápidamente a Java y ser productivos en poco tiempo.”⁷

- **Orientado a objetos.-** “Java es un lenguaje orientado a objetos, eso implica que su concepción es muy próxima a la forma de pensar humana”⁸.

Los objetos se agrupan en estructuras encapsuladas tanto sus datos como los métodos que manipula esa información.

- **Distribuido.** “Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como *http* y *ftp*. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales.”⁹
- **Robusto.** Java verifica el código al mismo tiempo que lo escribe, y otra antes de ejecutarse. Se realiza un descubrimiento de la mayor parte de los errores

⁶ <http://yosh35.tripod.com/id3.html>

⁷ <http://www.iec.csic.es/criptonomicon/java/quesjava.html>

⁸ <http://www.mailxmail.com/curso-java/caracteristicas-lenguaje-java-1>

⁹ <http://www.manual-java.com/manualjava/caracteristicas-java.html>

durante el tiempo de compilación, ya que Java verifica tipo de datos y declaraciones de variables.

“Respecto a la gestión de memoria, Java libera al programador del compromiso de tener que controlar especialmente la asignación que de ésta hace a sus necesidades específicas.”¹⁰

- **Seguro.** Al ejecutar el código java, realiza comprobaciones de seguridad, además el propio lenguaje carece de características inseguras.
- **Multiplataforma.** Es independiente de la plataforma en la que ha sido creada ya que la misma puede ser ejecutada en cualquier otra.
- **Portable.** La indiferencia a la arquitectura representa sólo una parte de su portabilidad. En Java todos los programas son iguales en todas las plataformas. Esto se debe gracias a la *Maquina Virtual Java* (JVM).
- **Multihebra.** Las aplicaciones que pueden ejecutar una acción a la vez se ven limitadas. Java es capaz de soportar sincronización de múltiples hilos de ejecución a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas.

“Mientras un hilo se encarga de la comunicación, otro puede interactuar con el usuario mientras otro presenta una animación en pantalla y otro realiza cálculos.”¹¹

1.7. JSP

“Es una tecnología que nos permite mezclar HTML estático con HTML generado dinámicamente.”¹²

¹⁰ http://pisuerga.inf.ubu.es/lsi/Invest/Java/Tuto/I_3.htm

¹¹ <http://www.iec.csic.es/cryptonicon/java/quesjava.html>

¹² http://www.programacion.com/java/tutorial/servlets_jsp/1/

JSP nos permite gestionar páginas web de una manera simplificada. Su objetivo es separar la interfaz (presentación visual) de la implementación (lógica de ejecución).

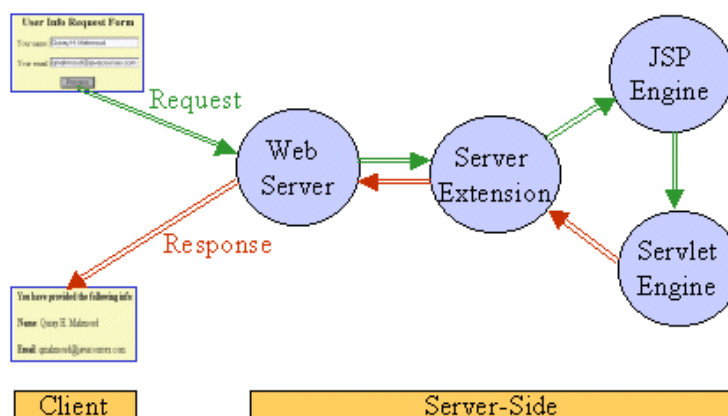


Figura 3. Funcionamiento del JSP

Fuente: <http://usabilidadweb.com.ar/jsp.php>

La principal ventaja es que permite integrarse con clases Java lo que permite separar en niveles las aplicaciones web, almacenando y el documento HTML en el archivo JSP. La idea fundamental es el de separar la lógica del negocio de la presentación de la información

1.7.1. MOTOR JSP

“El motor de las páginas JSP está basado en los servlets de aunque el número de desarrolladores que pueden afrontar la programación de JSP es mucho mayor, dado que resulta mucho más sencillo aprender que los servlets.”¹³

Los JSP son servlets pero de una manera simplificada los cuales actúan como tales después de la primera invocación, el motor se encarga de crear un .class para las futuras llamadas. Se utiliza los JSP por su manera simple ya que los servlets requieren un mayor conocimiento del Lenguaje Java

¹³ <http://www.desarrolloweb.com/articulos/831.php>

1.8. JSF

JSF nos ofrece una forma de trabajo que presta una facilidad al momento de desarrollar las diferentes aplicaciones web, separando el desarrollo en capas: presentación, reglas y entidades de negocio.

“JSF es un framework orientado a la interfaz gráfica de usuario (GUI), facilitando el desarrollo de éstas, y que sin embargo, realiza una separación entre comportamiento y presentación, además de proporcionar su propio servlet como controlador, implementando así los principios del patrón de diseño Model – view – Controller (MVC), lo que da como resultado un desarrollo más simple y una aplicación mejor estructurada”.¹⁴

Una de las características que JSF es que no está limitado a las características de HTML gracias a una gran variedad de renderes que tiene y el sin número de salidas que tiene para el usuario. Además de la adaptabilidad que tiene a la presentación que se lo hace mediante los JSP.

“Los JSF se integran dentro de la página JSP y se encarga de la recogida y generación de los valores de los elementos de la página.”¹⁵

JSF trata la vista o interfaz de usuario de una forma diferente a lo que se está familiarizado en aplicaciones web, ya que en esta forma se hace a través de componentes y está basada en eventos.

1.8.1. FUNCIONAMIENTO JSF

El funcionamiento de JSF es asociar la interfaz HTML con clases java, las que se encarga de procesar y recoger la información ingresada por el usuario.

Las aplicaciones JSF están formadas por los siguientes elementos principales:

¹⁴ http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/viveros_s_ca/capitulo3.pdf

¹⁵ <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=IntroduccionJSFJava>

- Páginas JSP que incluyen los formularios JSF. Estas páginas generarán las vistas de la aplicación.
- Beans java que se conectan con los formularios JSF.
- Clases java para la lógica de negocio y utilidades.
- Ficheros de configuración, componentes a medida y otros elementos del framework.

CAPITULO II

DETERMINACIÓN DE REQUERIMIENTOS Y ANÁLISIS

1.1. REQUERIMIENTOS BÁSICOS

El proceso realizado por el sistema de Georeferenciación de la ciudad de Quito, consiste en ubicar o encontrar un lugar, mediante información que el usuario ingresa, como por ejemplo una referencia, un lugar o una dirección, posibilitando que su búsqueda sea la deseada, en el menor tiempo y fácil de comprender.

Los tipos de datos utilizados por parte del sistema son direcciones o lugares los cuales se analizan, se procesan y se entregan al usuario a través de una representación grafica en un plano o croquis, la cual se relacionara a través de coordenadas con las imágenes satelitales que nos proveerá el sistema google EARTH.

La respuesta proporcionada por el sistema debe ser rápida y precisa, el sistema será capaz de funcionar en la web por lo cual cualquier usuario podría ingresar y obtener resultados del aplicativo.

Como todas las aplicaciones web se debe seguir los procesos de diseño, optimizando al máximo memoria y ancho de banda para que los usuarios no tengan inconvenientes, y puedan obtener los resultados deseados.

1.2. REQUERIMIENTOS FUNCIONALES

Los requerimientos con los que constara el sistema “Geosys” son:

Realizar consultas por nombre de distintas avenidas, lugares, barrios de la ciudad de Quito mostrando una imagen satelital clara y detallada, y sobrepuesto un croquis.

El sistema “Geosys ” está programado para los siguientes usuarios:

- Usuario Administrador
- Usuario Visitante
- Usuario Registrado

El Usuario Administrador podrá gestionar a los usuarios, a la información y la asignación de coordenadas que se maneja en el sistema.

Este usuario puede ingresar, actualizar, eliminar la información del sistema, esta puede corresponder a “gestión de usuario”, “gestión de información” y “gestión referente a ubicación”.

- Para la gestión de usuario podrá ingresar, editar, eliminar y buscar ya sea por nombre, ciudad o país.
- Para la gestión de información, tendrá el acceso de ingreso, asignaciones y consultas.
- En el ingreso de información se lo podrá realizar por zonas, cuadrantes, avenidas, lugares, barrios y tipos de lugares.
- Las asignaciones se manejaran mediante los cuadrantes con barrios, avenidas o haciendo una relación entre lugares y avenidas.
- Como último el administrador podrá realizar consultas de los datos ingresados por medio de avenidas, lugares y barrios.

Los Usuarios registrados tendrán mejores privilegios para la búsqueda requerida de una forma rápida, sencilla y además podrá realizar comentarios o sugerencias sobre el sitio Web.

Los usuarios visitantes tendrán una visión general de la funcionalidad del sistema “Geosys”, pero tendrán restricciones, pues no accederán a la imagen satelital.

1.3. REQUERIMIENTOS NO FUNCIONALES

Los requerimientos no funcionales para el sistema “Geosys” son los siguientes:

- El Sitio Web estará realizado bajo el lenguaje de programación JSF con conexión a Google Earth.
- El lenguaje que se utilizará para realizar el sitio web será Java Orientado a Objetos, con el entorno de desarrollo NetBeans 6.1.
- La estructura del programa será realizada a través de clases y paquetes.
- La base de datos del sitio Web será realizada en MySQL.
- En la aplicación del sistema se manejará el lenguaje de marcas KML basado sobre XML la cual será utilizada para la transmisión de datos con Google Earth.

1.3.1. REQUERIMIENTOS DEL SOFTWARE

Sistema Operativo: Multiplataforma, pero con las limitaciones propias del Google Earth con relación a los sistemas operativos

- Máquina Virtual de Java JVM
- Servidor web
- Google Earth Free

1.4. COMPRENSIÓN DEL PROCESO

Todos los días las personas deben prepararse para llegar, a ciertas direcciones, o lugares; por lo cual saber la ubicación precisa es de suma importancia, proporcionar un sistema que sea capaz de mostrar esta información ayudara para que todos puedan saber donde se encuentra, así no tendrá que usar información que en la mayoría de los casos son dificultosos de entender.

Generalmente para poder encontrar una dirección en la ciudad se procede a la búsqueda en un mapa impreso o guía, para luego asociar esa dirección con un cuadrante en el mapa. Tomando tiempo y en la mayoría de los casos encontrado varios resultados dificultando y tomando tiempo en la búsqueda.

El tiempo que generalmente se demora en realizar la búsqueda se lo plasma de manera muy simple por lo cual no se demora más que unos pocos segundos, dependiendo de la conexión de internet que el usuario disponga.

La mayoría de personas cuando tratan de obtener la dirección de un lugar en concreto toman la decisión de hacer la búsqueda por guías o por ayuda telefónica obteniendo de esta manera solo las calles de donde esta, sin ninguna otra referencia. Esto se realiza todos los días por personas de información (Ayuda telefónica) sin que los usuarios finales obtengan la ubicación precisa que necesitan.

1.5. IDENTIFICACIÓN DE LOS DATOS EMPLEADOS E INFORMACIÓN GENERADA

Uno de los pasos más importantes es detectar cuales son los datos empleados en el sistema, para ello se necesita la siguiente información: Barrios, Lugares como por ejemplo iglesias, parques, bibliotecas hospitales, además de calles o avenidas las cuales serán procesadas, interpretadas y finalmente comparada con la información contenida en una base de datos.

La información resultante es una imagen del plano de la ciudad de Quito mostrando la información que el usuario deseaba buscar, Además este croquis será traducido a un sistema de coordenadas gracias al cual se relacionara el croquis de la ciudad con la imagen satelital del sistema google Earth dando al usuario dos maneras diferentes de interpretación de la ubicación deseada.

1.6. FRECUENCIA Y VOLUMEN DEL PROCESO

El sistema de georeferenciación de la ciudad Quito es un sistema que estará al alcance de cualquier individuo que desee encontrar una dirección de una manera muy diferente a las convencionales, a este sistema se tendrá acceso todos los días del año, aunque no se puede determinar con exactitud una frecuencia y un volumen de proceso ya que esto dependerá mas del número de usuarios que visiten el sitio web, y en el momento en que se lo realice. Además del lugar y el tipo de información que se analice.

1.7. REQUERIMIENTOS DE LAS TRANSACCIONES DE LOS USUARIOS

Las transacciones que son realizadas por los usuarios, son de pedido, lo que significa que los usuarios realizaran peticiones al sistema el cual almacenara, interpreta los datos y los comparara para devolver una respuesta apropiada.

.

La transacción se realiza en el momento en que el usuario ingresa la información deseada, así que es quien la inicia y está haciendo el pedido de la información, con el propósito de encontrar, una ubicación.

El sistema está desarrollado para que todos los usuarios puedan realizar pedidos el número de veces que ellos los deseen, el volumen y la velocidad a la

que trabajan dependerá en gran manera del equipo utilizado y las características de navegación.

Los pedidos que se realicen dependerán de que se esté tratando de ubicar, y por esta característica el procesamiento de la información variara según sea un lugar, un barrio o una dirección, ya que la manera de procesar y adquirir los datos variara según sea el caso.

La gran mayoría de la información se encuentra almacenada en una base de datos, y relacionada con datos gráficos (Plano de la ciudad de Quito), y esta a su vez se acopla al sistema Google Earth. El cual nos facilita la visualización de las imágenes satelitales, relacionada con el croquis de la ciudad a través del sistema de coordenada, los datos de la ciudad puede ser actualizado de una manera constante por un usuario Administrador.

1.8. REQUERIMIENTOS DE DECISIÓN DE LOS USUARIOS

A los usuarios se les provee de tres maneras entre las que la puede examinar hasta encontrar la ubicación que este desee, la toma de esta decisión depende exclusivamente del beneficiario del sistema ya que este es el que tiene la información de la cual da inicio a las transacciones de la información.

La información inicial que debe tener el usuario, depende mucho de fuentes externas al sistema, ya que el aplicativo no la provee, esta información va referida a lo que este necesite localizar pudiéndola obtener de cualquier lugar o persona.

Los datos ingresados por el usuario son primeramente validados, y se procede a su verificación con datos anteriormente ingresados al sistema, estos datos que ya fueron verificados se los procesa para obtener la parte del plano de la ciudad y el sistema de coordenadas, produciendo la información necesaria para el usuario.

La información que el sistema necesita son datos debidamente organizados, si no se realiza este proceso el resultado final no será el que el usuario espera, si no una información no precisa, por lo cual dependerá del usuario formarla para su debido ingreso, además el sistema debe de contar con las debidas indicaciones, para no cometer errores y obtener los resultados deseados.

1.9. ANÁLISIS DEL SISTEMA

1.9.1. ORGANIZACIÓN

| Organización | UNIVERSIDAD POLITÉCNICA SALESIANA |
|--------------|---|
| Dirección | CAMPUS GIRÓN Avenida 12 de Octubre 1436 y Wilson CAMPUS KENEDY Rafael Bustamante s/n CAMPUS SUR Rumichaca y Morán Valverde s/n |
| Teléfono | (593) 2 2566327, (593) 2 2404190 |
| Fax | (593) 2 2626476 |
| Comentarios | Unidad de Educación Superior |

Tabla 1. Organización

La Universidad Politécnica Salesiana es la institución en la que el sistema será mostrado y evaluado, siendo la razón por la cual es indicada como la organización a la que los participantes y desarrolladores pertenecen.

1.9.2. SISTEMA

| Sistema | SYSGEO |
|-------------|---|
| Comentarios | Sistema de Georeferenciación de la ciudad de Quito Utilizando JSF y su conexión con Google Earth |

Tabla 2. Sistema

SysGeo: Sistema de Georeferenciación de la ciudad de Quito es el nombre del sistema a través del cual podremos ayudarnos al momento de tratar de encontrar una dirección o lugar en la ciudad.

1.9.3. PARTICIPANTES

| | |
|---------------------|-----------------------------------|
| Participante | EDISON JUMBO |
| Organización | Universidad Politécnica Salesiana |
| Rol | Programador |
| Es desarrollador | Sí |
| Comentarios | Desarrollador del sistema |
| Participante | JOHANNA ANILOA |
| Organización | Universidad Politécnica Salesiana |
| Rol | Programador |
| Es desarrollador | Sí |
| Comentarios | Desarrollador del sistema |
| Participante | ING. GUSTAVO NAVAS |
| Organización | Universidad Politécnica Salesiana |
| Rol | Tutor |
| Comentarios | Docente guía del proyecto |

Tabla 3. Participantes en el desarrollo del sistema

Las dos personas encargadas de desarrollar el proyecto de georeferenciación de la ciudad de Quito son los dos primeros participantes. Los cuales tendrán que realizar la investigación y posterior creación del sistema, seguido de las

indicaciones y pautas que se darán por el Ing. Gustavo Navas como director del proyecto.

1.9.4. ACTORES.

| ACT-0004 | VISITANTE |
|-----------------|---|
| Versión | 1.0 (24/01/2010) |
| Autores | EDISON JUMBO JOHANNA ANILOA |
| Descripción | Este actor representa <i>a las personas sin registrarse</i> |
| Comentarios | Este tipo de usuario no necesita registrarse y tiene pocos beneficios del sistema |
| Revisado por: | ING. GUSTAVO NAVAS |

Tabla 4. Autor Visitante

Este usuario del sistema no necesita estar registrado y obtendrá pocos beneficios. El sistema no realizará la conexión entre el aplicativo y Google Earth, no pudiendo de esta manera realizar la comparación entre el plano de la ciudad y la fotografía satelital.

| ACT-0005 | REGISTRADO |
|-----------------|--|
| Versión | 1.0 (24/01/2010) |
| Autores | EDISON JUMBO JOHANNA ANILOA |
| Descripción | Este actor representa <i>a los clientes logeados y con acceso a la información</i> |
| Comentarios | Son los usuarios que pueden ingresar al sistema y realizar búsquedas de una manera específica. |
| Revisado por: | ING. GUSTAVO NAVAS |

Tabla 5. Autor Registrado

Este tipo de usuario podrá registrarse y además tendrá acceso a toda la información que el sistema tiene almacenado haciendo comparaciones y

búsquedas de una manera interactiva pudiendo obtener de esta manera datos más detallados, y podrá dejar comentarios o criticas referente al sistema y su funcionamiento.

| ACT-0006 | ADMINISTRADOR |
|-----------------|---|
| Versión | 1.0 (24/01/2010) |
| Autores | EDISON JUMBO JOHANNA ANILOA |
| Descripción | Este actor representa <i>a las personas que administran la información del sistema</i> |
| Comentarios | Estas personas pueden realizar cualquier tipo de tratamiento con la información del sistema |
| Revisado por: | ING. GUSTAVO NAVAS |

Tabla 6. Autor Administrador

El usuario administrador es quien ingresa la información, la puede modificar e inclusive eliminar si no cumple con los requerimientos del sistema. Además puede manipular toda la información correspondiente a los usuarios registrados, como sus comentarios o sugerencias. Este usuario también podrá actualizar e inclusive eliminar a un usuario del sistema.

1.9.5. CASOS DE USO

| UC-0001 | GESTION DE INFORMACION | |
|------------------|--|---|
| Versión | 1.0 (24/01/2010) | |
| Autores | EDISON JUMBO | |
| | JOHANNA ANILOA | |
| Descripción | El sistema deberá comportarse tal como se describe en el siguiente caso de uso | |
| Precondición | Los usuarios deben estar registrados | |
| Secuencia normal | Paso | Acción |
| | 1 | El actor ADMINISTRADOR (ACT-0006) <i>ingresa de información vital del sistema</i> |

| | | |
|---------------------|---|--|
| | 2 | El actor ADMINISTRADOR (ACT-0006) <i>actualiza la información de la base de datos</i> |
| | 3 | El actor ADMINISTRADOR (ACT-0006) <i>elimina información deficiente de la base de datos</i> |
| | 4 | El actor ADMINISTRADOR (ACT-0006) <i>consulta datos con especificaciones</i> |
| | 5 | El actor REGISTRADO (ACT-0005) <i>consulta datos con especificaciones</i> |
| | 6 | El actor VISITANTE (ACT-0004) <i>consulta datos generales</i> |
| | 7 | <i>Si el administrador genera información, el sistema se conectara a la base de datos y se almacenaran los datos.</i> |
| Postcondición | Información almacenada en una base de datos | |
| Excepciones | Paso | Acción |
| | 3 | <i>Si los datos no son asignados a coordenadas se eliminan, el actor ADMINISTRADOR (ACT-0006) podrá borrar la información que no esté con coordenadas relacionadas, a continuación este caso de uso queda sin efecto</i> |
| | 7 | <i>Si la información está mal ingresada, el sistema enviara un mensaje al usuario y, a continuación este caso de uso queda sin efecto</i> |
| Frecuencia esperada | 100 veces por día(s) | |
| Importancia | Vital | |
| Urgencia | Inmediatamente | |
| Estado | Validado | |
| Estabilidad | Alta | |

| | |
|---------------|--|
| Comentarios | Es de vital importancia ya que con este modulo se administra e interactúa con la información que permite el funcionamiento del sistema |
| Revisado por: | ING. GUSTAVO NAVAS |

Tabla 7. Caso de Uso Gestión de Información

Este es el caso de uso más importante ya que gracias a él se puede tener acceso a la información con la cual el sistema puede funcionar, además se puede especificar cuáles son las características de la información y a qué grupo pertenece.

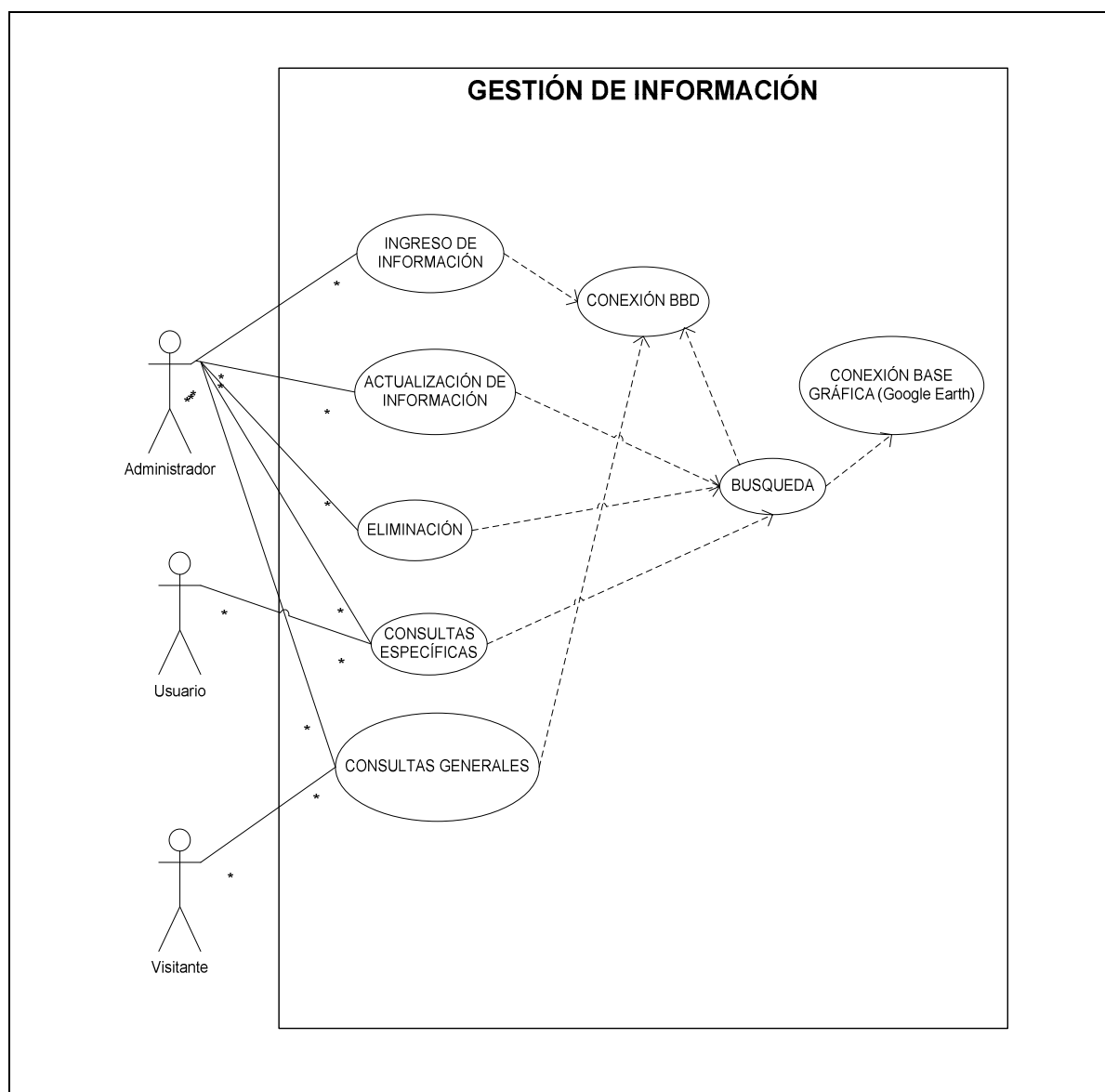


Figura 4. Caso de Uso Gestión de Información

Fuente: El Autor

| | | |
|---------------------|--|--|
| UC-0002 | GESTION DE USUARIOS | |
| Versión | 1.0 (24/01/2010) | |
| Autores | EDISON JUMBO JOHANNA ANILOA | |
| Descripción | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando sea el <i>ingreso al sistema</i> | |
| Precondición | Disposición del hardware y software para el sistema | |
| Secuencia normal | Paso | Acción |
| | 1 | El actor ADMINISTRADOR (ACT-0006) se <i>registra</i> |
| | 2 | El actor REGISTRADO (ACT-0005) <i>se registra</i> |
| | 3 | El actor ADMINISTRADOR (ACT-0006) <i>registra, actualiza, elimina y obtiene reportes</i> |
| | 4 | El actor VISITANTE (ACT-0004) <i>se puede registrar</i> |
| Postcondición | Usuario registrado y sus datos almacenados en la base de datos | |
| Excepciones | Paso | Acción |
| | 4 | <i>si la información requerida no está detallada, el sistema anula el registro y, a continuación este caso de uso queda sin efecto</i> |
| Frecuencia esperada | 100 veces por día(s) | |
| Importancia | Vital | |
| Urgencia | hay presión | |
| Estado | Validado | |
| Estabilidad | Alta | |
| Revisado por: | ING. GUSTAVO NAVAS | |

Tabla 8. Caso de Uso Gestión de usuarios

Es este caso de uso se especifica que actividades se pueden realizar con los usuarios tanto registrados, no registrado y los administradores del sistema, como se realiza su registro y el tratamiento de su respectiva información

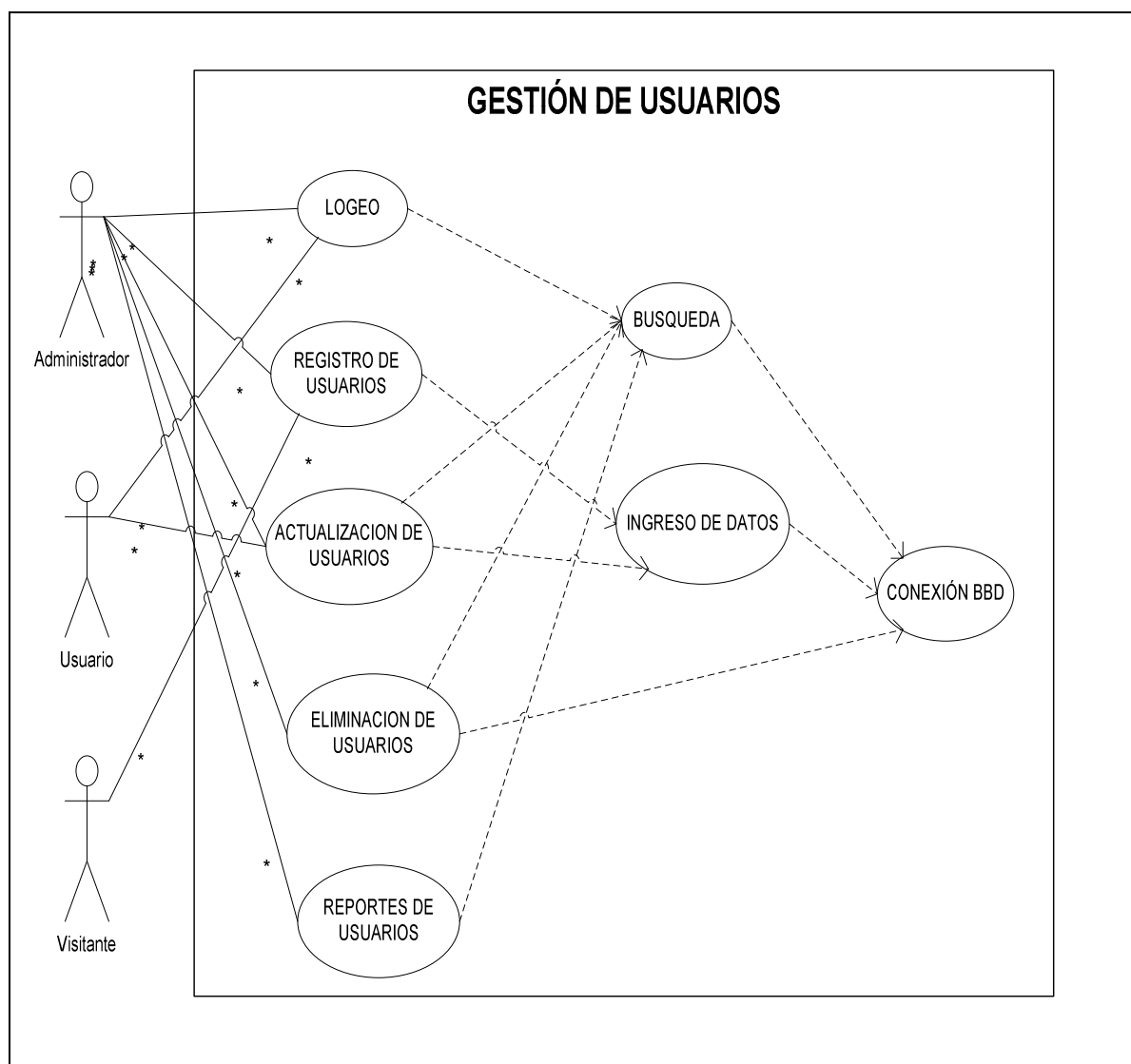


Figura 5. Caso de Uso Gestión de Usuarios

Fuente: El Autor

| UC-0003 | GESTION DE SUGERENCIAS | |
|---------------------|--|---|
| Versión | 1.0 (24/01/2010) | |
| Autores | EDISON JUMBO JOHANNA ANILOA | |
| Dependencias | Ninguna | |
| Descripción | El sistema deberá comportarse tal como se describe en el siguiente caso de uso abstracto durante la realización de los siguientes casos. | |
| Precondición | Los usuarios deben estar registrados | |
| Secuencia normal | Paso | Acción |
| | 1 | El actor REGISTRADO (ACT-0005) <i>ingresa la sugerencia</i> |
| | 2 | El actor ADMINISTRADOR (ACT-0006) <i>revisa, toma y elimina la sugerencia</i> |
| Postcondición | La información es almacenada en una base de datos | |
| Frecuencia esperada | 100 veces por mes(es) | |
| Importancia | Importante | |
| Urgencia | puede esperar | |
| Estado | pendiente de validación | |
| Estabilidad | Media | |
| Comentarios | Es importante ya que podemos retroalimentar la información y mejorar la aplicación. | |
| Revisado por: | ING. GUSTAVO NAVAS | |

Tabla 9. Caso de Uso Gestión de sugerencias y comentarios

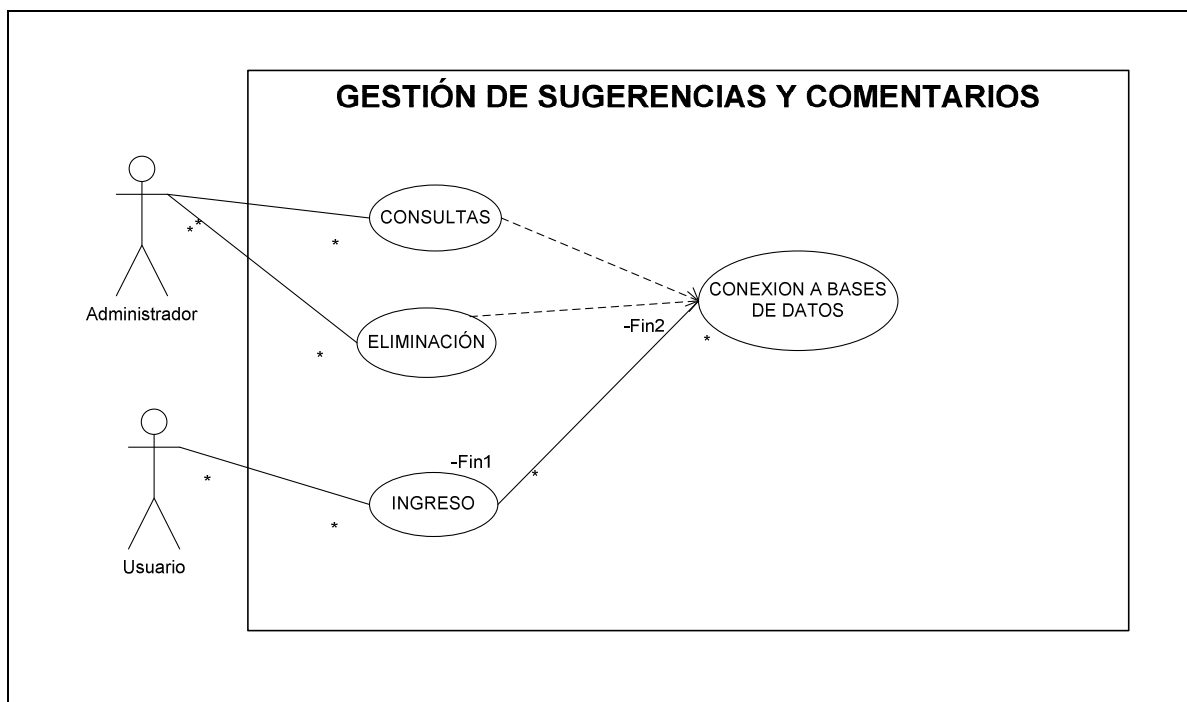


Figura 6. Caso de Uso Gestión de sugerencias y comentarios

Fuente: El Autor

Con este caso de uso podemos administrar las sugerencias y comentarios, y retroalimentar nuestra interfaz, para de esta manera estar al tanto de las necesidades y requerimientos de los usuarios.

Para el desarrollo del análisis de sistema se utilizó el sistema REM The Free Requirements Management Tool, creador por Amador Durán Toro el cual especifica en la licencia que tipo de software el cual se detalla a continuación.

“REM es una herramienta CASE experimental para la ingeniería de requisitos. Está diseñada para uso académico por lo que su uso para fines comerciales no se permite sin permisos por parte del autor”¹⁶.

¹⁶ . Amador Durán Toro Creador del sistema y docente de la Universidad de Sevilla.

CAPÍTULO III

DISEÑO DEL SISTEMA

3.1. CONCEPTOS Y PRINCIPIOS

Con el desarrollo del Análisis se obtuvo una gran variedad de información que ayudara en el diseño del sistema. Con esta información se puede saber el funcionamiento del sistema y de los subsistemas que este posee, con suficientes detalles como para permitir su interpretación y realización física.

El sistema SysGeo trabaja con la tecnología java y por capas, dándonos una capacidad de diseño mucho mayor. El sistema contara con un número de clases que ayudaran: en la asignación de valores que necesitan las herramientas graficas (Interfaces de usuario), las conexiones a la base de datos y futura creación de los archivos KML (Archivos necesarios para la comunicación entre el aplicativo y Google Earth).

La tecnología con la que cuenta el sistema es JSF, la que permite hacer una separación por capas (interfaces, conexión y lógica de negocios). Por lo cual el sistema contara con varias clases las cuales nos ayudaran a lograr un mejor desempeño del aplicativo y superior forma de diseño

3.2. EL DISEÑO DE LOS DATOS.

Los datos que el sistema necesita para poder trabajar de una manera segura y efectiva serán ingresados por los diversos usuarios, este ingreso se deberá hacer por pantalla, muchos de los datos son cadenas de textos los cuales serán procesados y almacenados para poder mostrar el resultado final al usuario.



3.3. CLASES

Figura 7. Diagrama de clases autor visitante
Fuente: El Autor

3.3.1. CLASES

El aplicativo sysgeo está compuesto por varias clases las cuales realizan una tarea específica para un usuario determinado, y estos solo tienen acceso a las clases previamente definidas para cada uno.

3.3.1.1. CLASES FRAGMENTO

JSF incorpora la posibilidad de fragmentar una página, estos fragmentos son interfaces independientes con sus respectivas clases los cuales pueden ser agregados a cualquier otra que uno desee, facilitando el diseño del sistema.

Se pueden realizar estos fragmentos para realizar banners, pies de página e inclusive menús. Todas las interfaces del aplicativo tienen varias fragmentaciones diferentes para cada tipo de usuario que se registre.

| CLASE FRAGMENTO | DESCRIPCIÓN |
|--------------------|---|
| Banner | Es el encabezado de todas las paginas que existen en el aplicativo |
| Menu | Es el menú del usuario invitado el cual le guiara a las interfaces que tiene acceso como no registrado |
| Menú_administrador | Menú del usuario administrador nos muestra los enlaces para el respectivo trabajo que este usuario tiene. |
| Menu_usuario | Muestra las guía hacia las interfaces que pueden ayudar a los usuarios registrados en sus búsquedas respectivas |
| Login | Este fragmento se mostrara en las clases que son de uso específico para los usuarios no registrados o en la interfaz de bienvenida en el sistema. |

| | |
|-----|---|
| Pie | Al igual que el banner el fragmento se muestra en todas las interfaces mostrando información extra del sistema y sus autores. |
|-----|---|

Tabla 10. Clases Fragmentos usadas en el sistema

3.3.1.2. CLASES DE USO DEL USUARIO VISITANTE

No todos los tipos de usuarios tienen acceso a la información completa o son capaces de modificarla por lo cual los usuarios se deben registrar para el uso de los datos y la manipulación de los mismos.

A continuación se detallan las clases que el usuario visitante tiene acceso con sus respectivas interfaces.

| CLASE | DESCRIPCIÓN |
|--------------------|---|
| Page1 | Página Principal del sistema es la encargada de dar la bienvenida al usuario y a la cual todos tienen acceso, y también brinda los enlaces para los demás tipos de usuario y las características que esta brinda. |
| Registro | Clase que permite registro de nuevos usuarios, los cuales podrán registrarse con un mínimo de información y tener acceso a varias ventajas que brinda el aplicativo. |
| Busqueda_usuarios2 | Es una búsqueda limitada que los usuarios visitantes pueden realizar, solo interactúa con el plano de la ciudad de Quito con características propias de él. |
| Sugerir | Permite al usuario invitado ingresar sugerencias para de esta manera pueda indicar mejoras o errores del sistema para futuras correcciones. |

Tabla 11. Clases de uso del usuario visitante



Figura 8. Clases usuario registrado
Fuente: El Autor

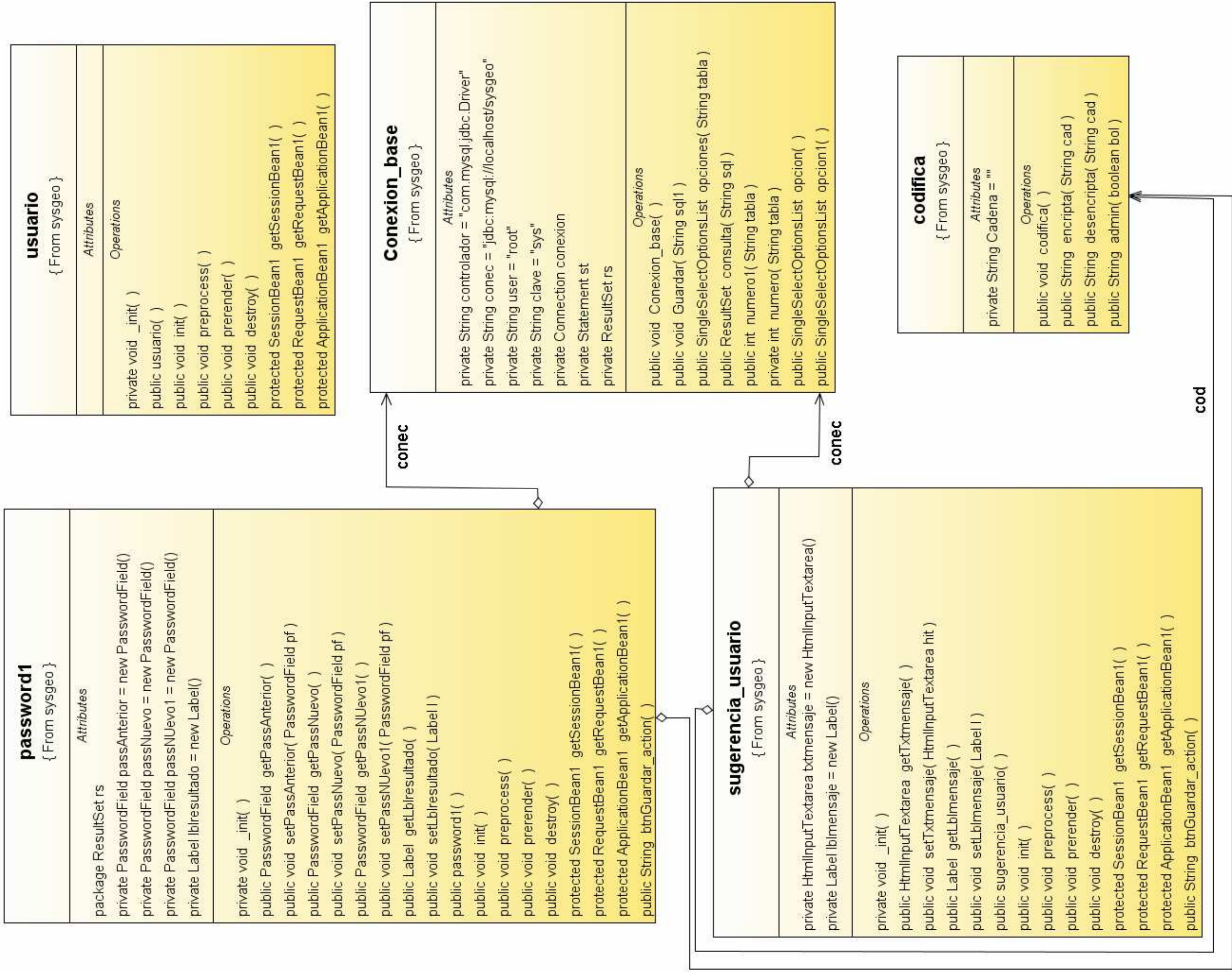


Figura 9.
Administración datos
del usuario registrado
Fuente: el autor

3.3.1.3. CLASES DE USO DEL USUARIO REGISTRADO

El usuario registrado tiene acceso a información y presentación de la misma de una manera más amplia

| CLASE | DESCRIPCIÓN |
|---------------------|---|
| Usuario | Página bienvenida del usuario registrado |
| Busqueda_usuario1 | Búsqueda general del usuario registrado con todas las características, datos y con la conexión a la base gráfica de Google earth. |
| Sugerencias_usuario | Permite a los usuarios registrado realizar sugerencias sobre el sistema, para correcciones o implementaciones de mejoras en el aplicativo |
| Password1 | Permite a los usuarios registrado realizar el cambio de contraseña de su cuenta. |

Tabla 12. Clases de uso del usuario registrado



Figura 10. Búsqueda del usuario registrado

Fuente: El Autor

3.3.1.4. CLASES DE USO DEL USUARIO ADMINISTRADOR

| CLASE | DESCRIPCIÓN |
|-------------------------|--|
| Administrador | La clase administrador está asociada con la interfaz de bienvenida, la cual tendrá los fragmentos que están asociados a este tipo de usuario y los enlaces respectivos a las tareas de este. |
| Admin_Contra | Esta clase permite que el tipo de usuario administrador pueda cambiar su contraseña y de esta manera pueda mantener la seguridad en los procesos que este maneja. |
| Asig_cuadrante_avenida | Una vez que han sido ingresados los cuadrantes y avenidas se procede a hacer una asociación entre estas dos para poder obtener una coordenada de la calle ingresada. |
| Asig_cuadrante_barrio | Los cuadrantes se podrán asociar con los barrios por esta clase. |
| Asig_lugares_cuadrantes | Los lugares ya ingresados se asocian con las avenidas para su localización y mayor exactitud. |
| Búsqueda_g | Se puede administrar a todos los usuarios y además cambiar el tipo de usuario al que pertenece. |
| Ing_avenidas | La mayor fuente de información es ingresada y procesada por esta clase ya que la mayoría de las búsquedas se realizarán por este parámetro. |
| Ing_barrios | Pide la información de los barrios de la ciudad para que se puedan administrar y para su futura localización. |

| | |
|------------------|---|
| Ing_cuadrantes | Los cuadrantes son los que se relacionan directamente con las coordenadas y estas a su vez deben relacionarse con el resto de la información para poder localizarla en las imágenes satelitales. |
| Ing_informacion | Ingreso de zonas |
| Ing_lugares | Con esta clase se ingresa y administra los lugares más conocidos de la ciudad. |
| Ing_tipo_lugar | Por el tamaño y estructura de la ciudad con esta clase se especifica qué tipo de lugar como un hospital, museo, etc. |
| Ingre_user | Otra forma de poder crear una cuenta para el uso del aplicativo es a través del administrador dándole características de administrador o solo usuario registrado. |
| Busqueda_usuario | Búsqueda general del usuario administrador con todas las características, datos y con la conexión a la base gráfica de google earth, además este usuario permite que los usuarios administrador confirmar las repuestas de datos. |

Tabla 13. Clases de uso del usuario administrador

Los diagramas de ingreso de datos son similares al anterior conectado a las mismas clases.

3.3.1.5. CLASES DE USO DE USO COMÚN

Estas clases son utilizadas por todos los usuarios del aplicativo de manera indirecta, a través de las clases propias que ellos utilizan ayudando a obtener datos, guardar, crear archivos, llenando objetos por ejemplo tablas que muestra la información del sistema.

- **Archivos:**

A través de la clase archivos se pueden crear los archivos KML que son los que trabajan con Google Earth dándole como referencia las coordenadas y el nombre del cuadrante donde se encuentra la información necesitada por el usuario.

- **Codifica**

Las claves de ingreso pueden ser obtenidas por medios diferentes al que el sistema provee, para evitar esto se codifican las claves de ingreso e información importante para los usuarios, además ayuda en la comparación de datos cifrados.

- **Conexión_base:**

La clase principal que ayuda al ingreso de datos y tratamiento de la información, es la que permite el funcionamiento del sistema al almacenar y consultar la información que los usuarios necesitan, además permite que los objetos del sistema se llenen de datos que son importantes para el usuario.

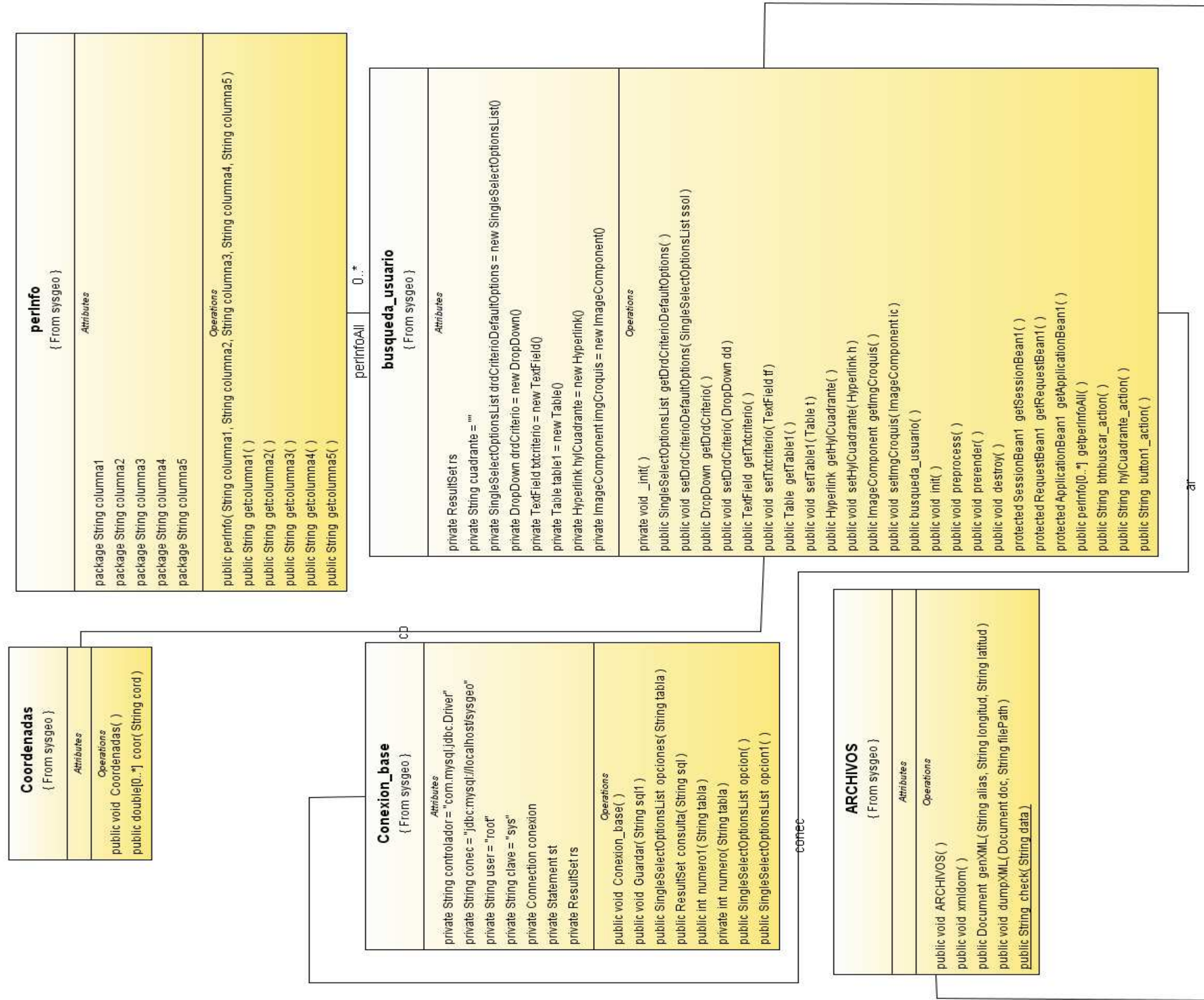


Figura 13. Clases de uso Común y Búsqueda de datos Fuente: El autor

- **Coordenadas:**

Los datos están asociados con cuadrantes, los cuadrantes se asocian con las coordenadas (latitud, longitud), la clase encargada de realizar este cálculo es la clase coordenadas. Obteniendo los datos a través de cálculos matemáticos y punto de referencia de la ciudad.

- **PerInfo:**

La clase perInfo es una de las clases más importantes ya que permite que las tablas de datos que están en la aplicación puedan mostrar su contenido, de esta manera es como los datos se asigna a su respectiva columna y como obtiene su valor. La clase está diseñada de tal manera que puede llenar cualquier tabla con cualquier valor

- **Usuarios:**

Al igual que la clase perInfo la clase usuarios es la encargada de mostrar la información en tablas pero de una manera más detallada.

3.4. PRINCIPALES TAREAS DEL SISTEMA

El presente proyecto tiene que realizar varias tareas para que su funcionamiento sea el correcto y para poder mostrar lo resultados requeridos por cada usuario:

- Registro
- Autenticación (Logeo)
- Ingreso de información
- Eliminación de información
- Actualización de información

3.4.1. REGISTRO DE USUARIOS.

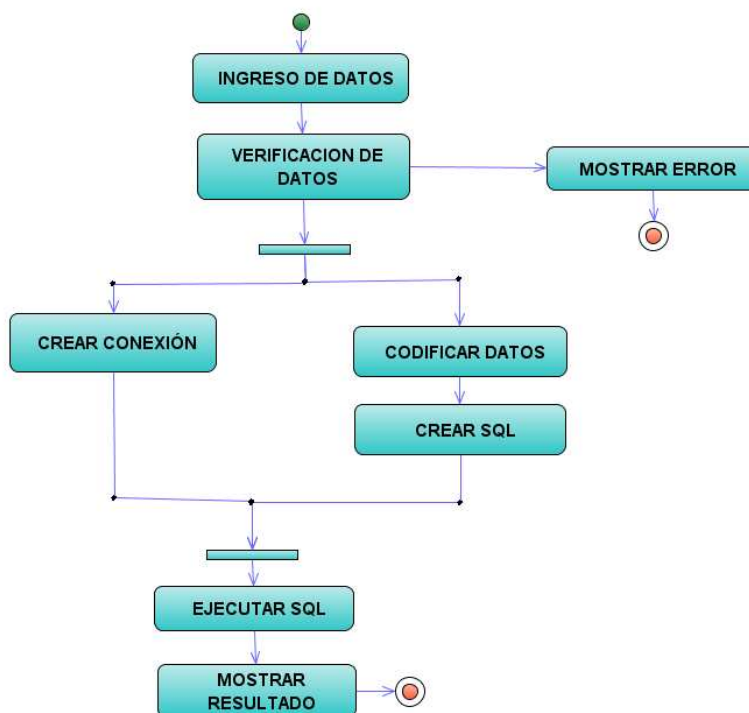


Figura 14. Diagrama de actividad registro de Usuarios

Fuente: El Autor

El primer paso que todo usuario debe realizar para utilizar el sistema es registrarse. El registro es totalmente libre y voluntario, los datos suministrados por las personas son básico solo para identificar los usuarios que utilizan el sistema, el proceso que se realiza para el almacenamiento de estos se explican en la Figura N°. 14.

El sistema reconoce 3 tipos de usuarios, invitado, registrado, y administrador con el registro solo se puede obtener los beneficios del segundo usuario, solo un usuario administrador puede dar beneficios equivalente a los suyos a otro usuario.

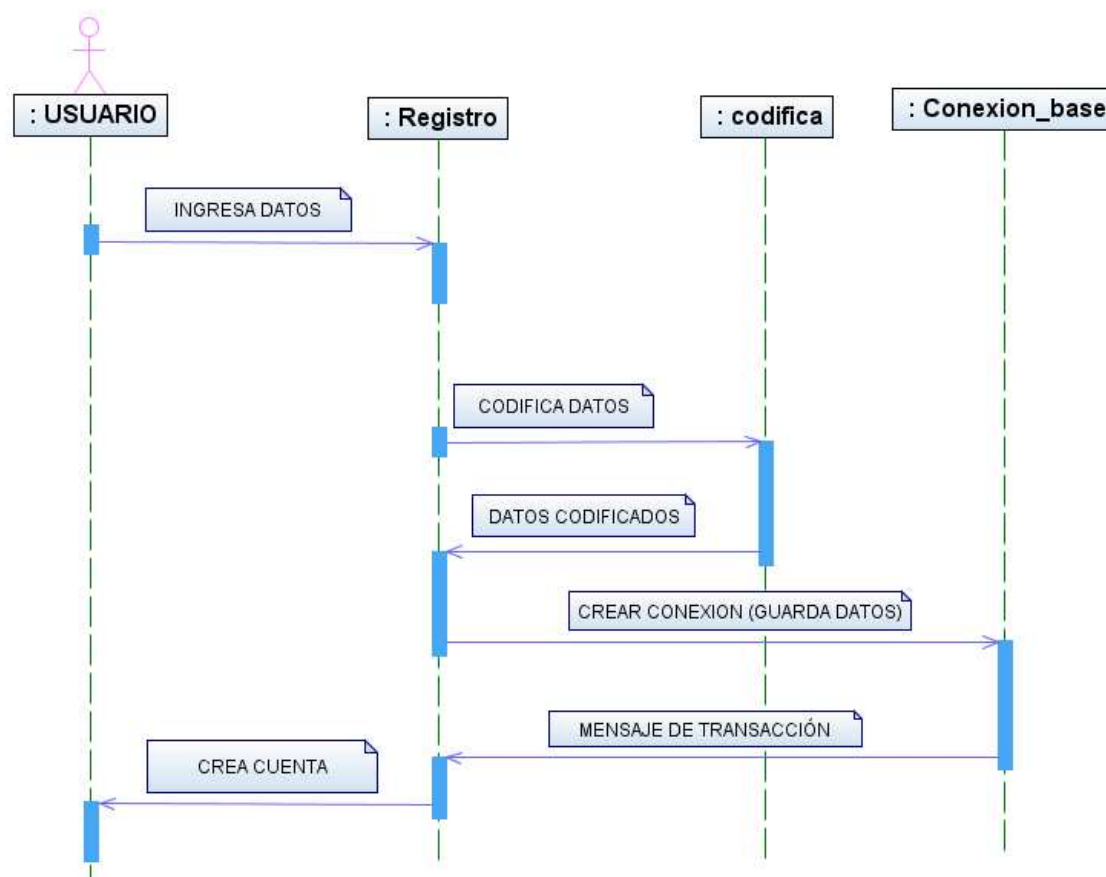


Figura 15. Diagrama de secuencia registro de Usuarios

Fuente: El Autor

Para registrarse se debe ejecutar diversas tareas, las cuales son realizadas por los objetos y las clases especializadas en este trabajo, como codificar los datos, crear la conexión con la base de datos y ejecutar las sentencias SQL: cuyo proceso se explica en la Figura N°. 15

3.4.2. AUTENTICACIÓN DE USUARIOS

Cada usuario que se ha registrado en el sistema debe autenticarse para poder obtener los resultados deseados. Si esto no se realiza el usuario obtendrá acceso a datos limitados. El proceso de autenticación es simple se ingresa un Nick y una Password de ingreso si los datos son los correctos se crea una sesión con sus

respectivos permisos a sus interfaces correspondientes como se explica en la Figura N°. 16

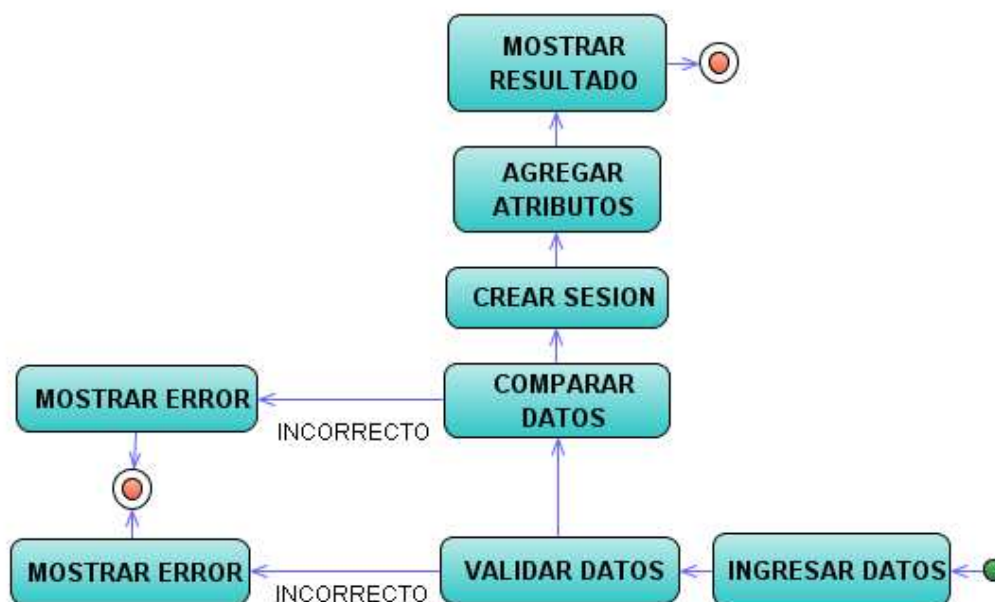


Figura 16. Diagrama de actividad de Autenticación de usuarios

Fuente: el Autor

Las clases empleadas y los objetos en este proceso son aquellas que codifican, decodifican y la conexión a las bases de datos como se indica en la Figura N°. 11. Luego de la cual se procede a crear la sesión con sus atributos y a enviar a la interfaz de respuesta para cada usuario.

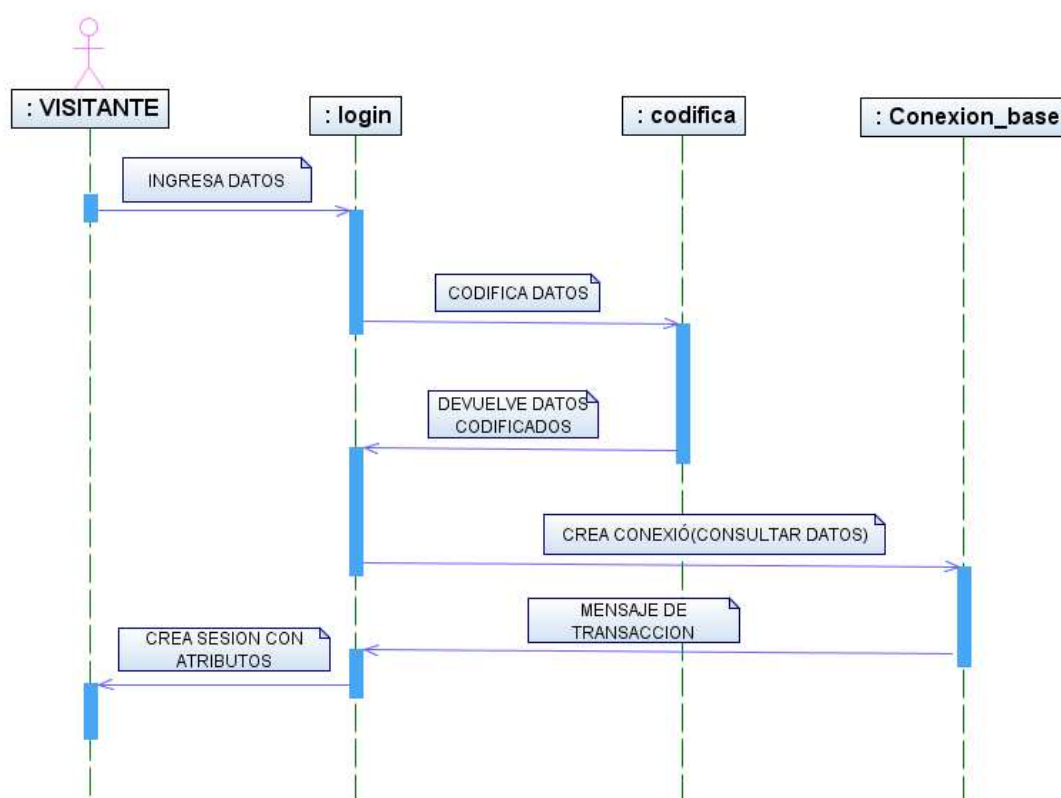


Figura 17. Diagrama de secuencia de Autenticación de usuarios

Fuente: El Autor

3.4.3. INGRESO DE INFORMACIÓN

El ingreso de información se refiere al almacenamiento de datos los cuales serán empleados directamente por el sistema estos datos son calles, lugares y barrios los cuales serán procesados para un futuro resultado.

El proceso de almacenar la información se lo realiza según la Figura N°. 19. El cual muestra como se verifica, almacena y muestro el resultado del proceso de almacenamiento de datos.

Además en la Figura N°. 18. Muestra los objetos y clases que intervienen y los mensajes de interacción entre estos permitiendo la conexión y almacenamiento de datos.

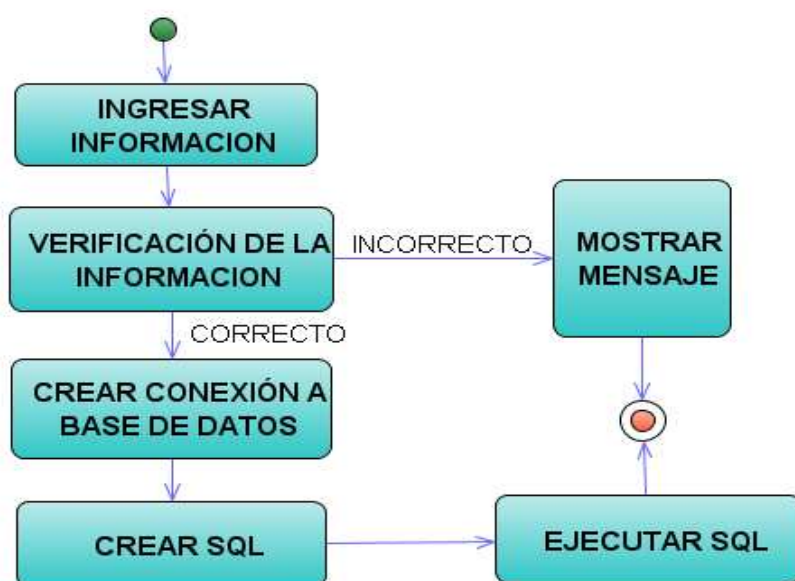


Figura 18. Diagrama de actividad ingreso de información

Fuente: El Autor

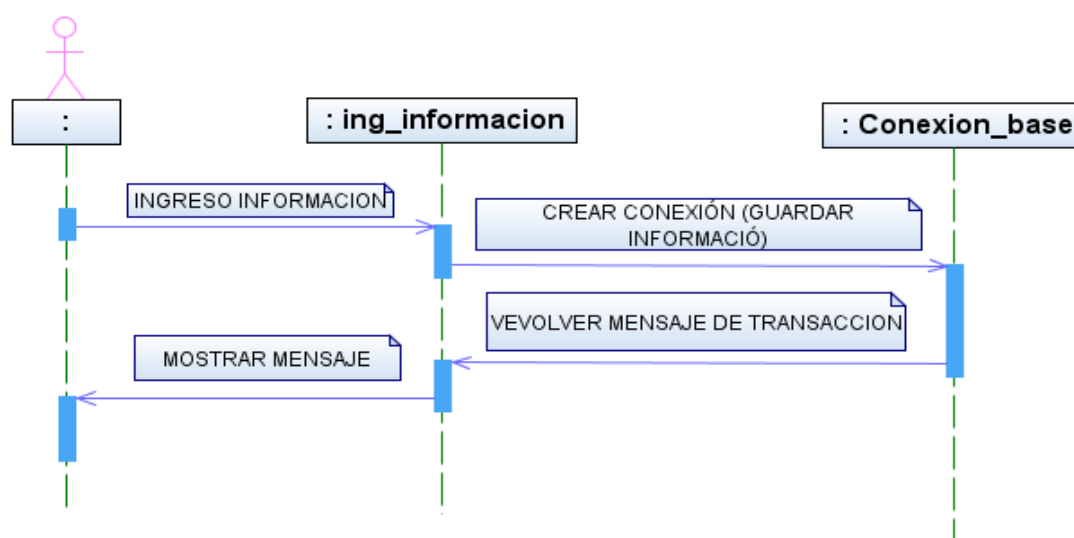


Figura 19. Diagrama de secuencia ingreso de información

Fuente: El Autor

3.4.4. TRATAMIENTO DE LA INFORMACIÓN

No todos los datos ingresados son correctos o están listos para ser usados estos datos deben de ser revisados para futura corrección o eliminación. Los datos deben ser revisados si estos tienen errores como por ejemplo faltas ortográficos

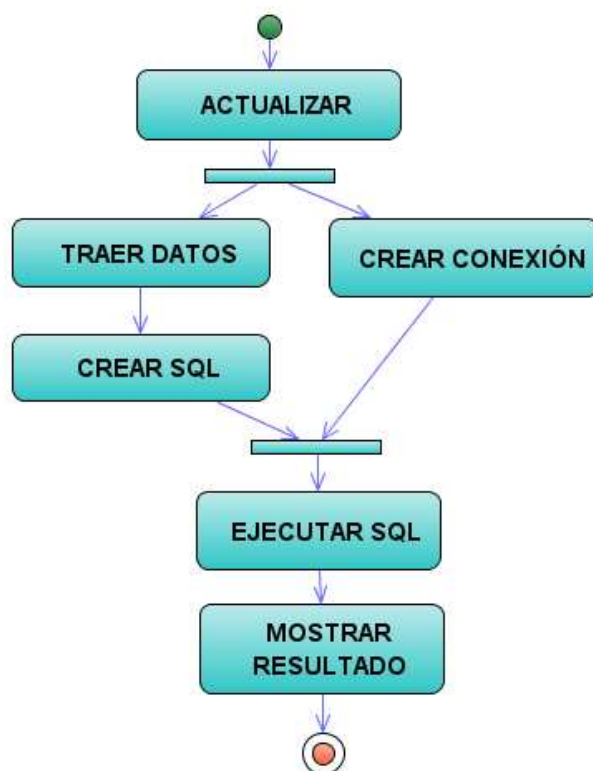


Figura 20. Diagrama de actividad actualización de información

Fuente: El Autor

estos deben ser actualizados como la Figura N°. 20. indica que proceso se debe realizar.

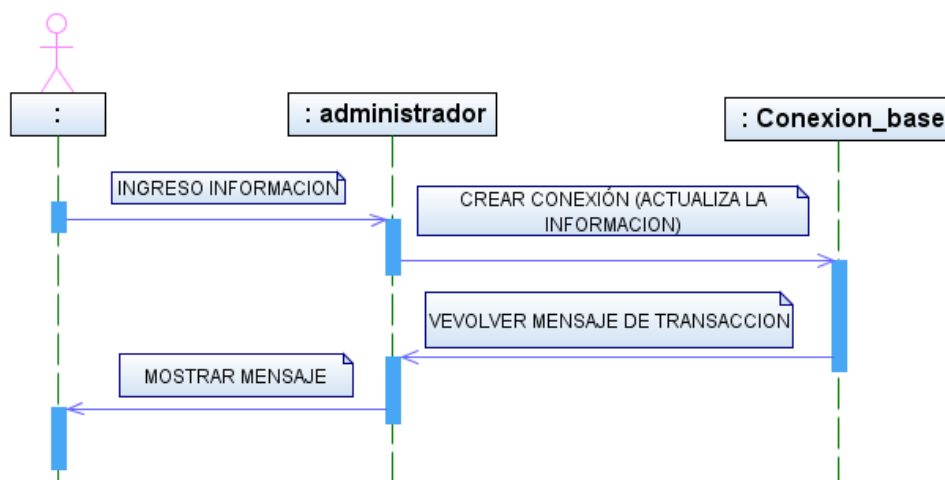


Figura 21. Diagrama de secuencia actualización de información

Fuente: el autor

Para actualizar los datos si estos necesitan codificación se debe realizar y utilizar clases y objetos necesarios para dicho proceso de almacenamiento o actualización de los datos como se explica en la Figura N°. 22.

Si los datos contienen errores los cuales no pueden ser corregidos se necesita de

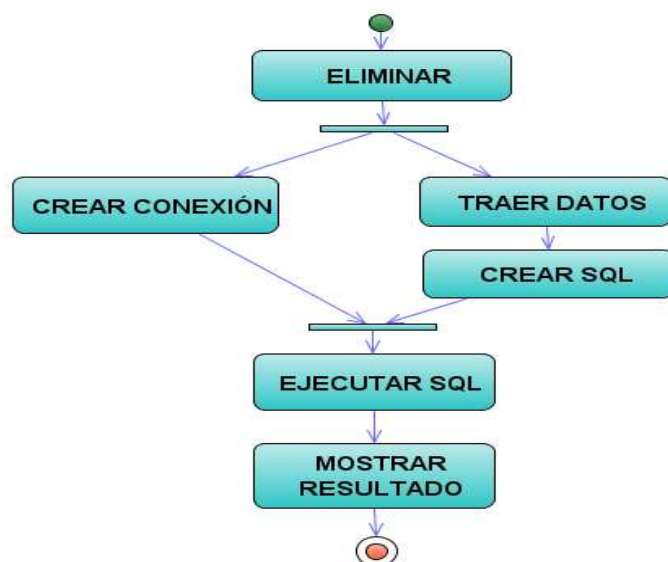


Figura 22. Diagrama de actividad eliminación de información

Fuente: El Autor

la eliminación para cuyo proceso se necesita un identificativo o código con el cual se inicia esta tarea Figura N°. 22.

Los objetos y clases que intervienen en la eliminación de la información y cómo interactúan entre ellas se explica de una manera simple en la Figura N°. 23. En el cual se muestra con la facilidad que se realiza el borrado de datos solo con la conexión a la base de datos.

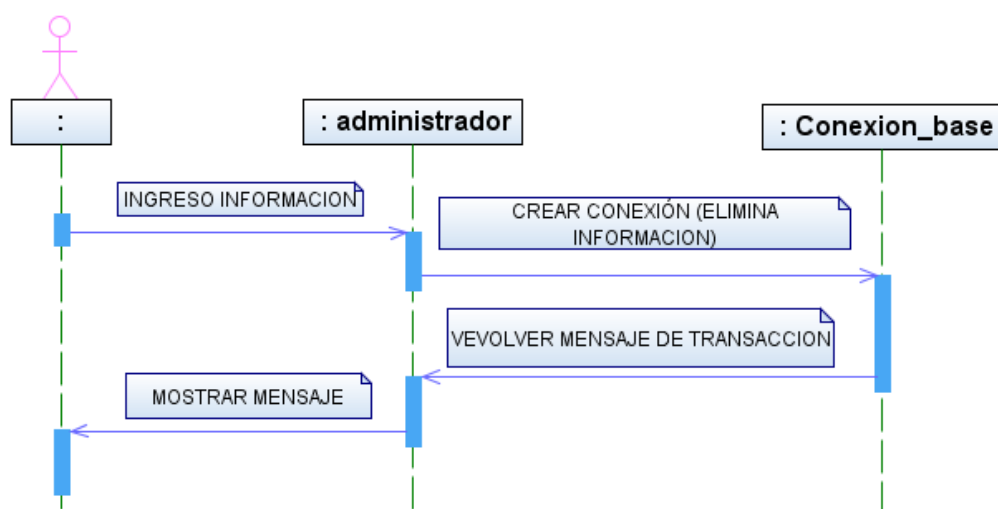


Figura 23. Diagrama de secuencia eliminación de información

Fuente: El Autor

3.4.5. ASIGNACIÓN DE INFORMACIÓN

La asignación de información se refiere a relacionar los datos como por ejemplo un lugar con un cuadrante, esta tarea tiene que ser realizada con toda la información que está ingresada para que de esta manera sea procesada y utilizada por el usuario final del sistema.

Este proceso radica su importancia ya que con este se puede obtener una coordenada de una búsqueda requerida por un usuario. El proceso por el cual se realiza se explica en siguiente Figura N°. 24.

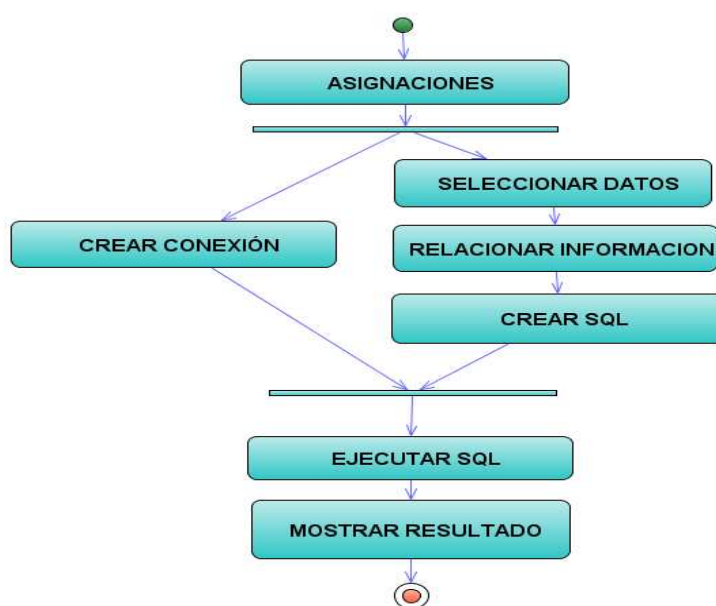


Figura 24. Diagrama de actividad asignación de información Fuente: El Autor

Las clases y las interfaces que se utilicen para este proceso variaran según qué datos se deseen relacionar y como se lo va hacer.

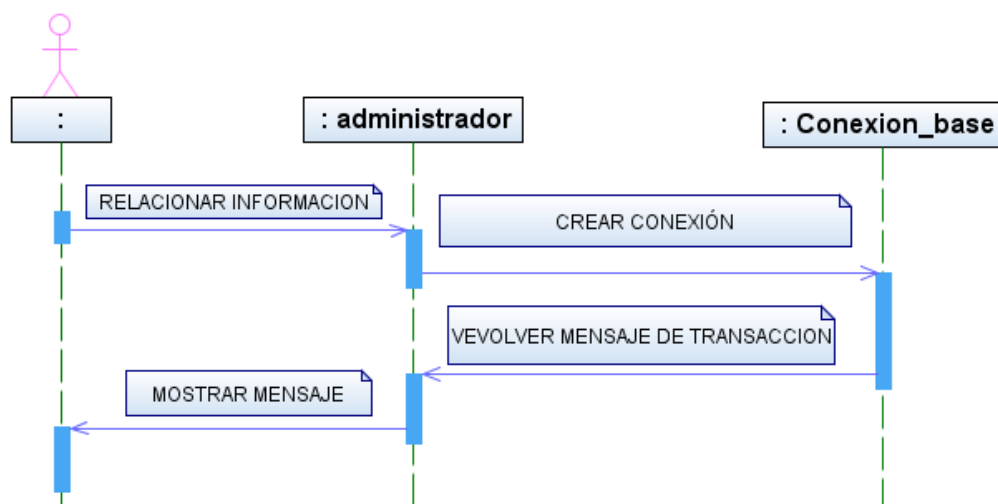


Figura 25. Diagrama de actividad asignación de información Fuente: El Autor

3.4.6. BÚSQUEDA DE INFORMACIÓN

Existen dos tipos de búsquedas utilizando este sistema. Entre los cuales están el usuario invitado que tiene una búsqueda sin privilegios y los usuarios registrados los cuales tiene búsqueda con privilegios.

La diferencia entre estas dos búsquedas se encuentra en que la que no tiene privilegios solo puede interactuar con el croquis de la ciudad de Quito; mientras que la búsqueda con privilegios puede interactuar con este croquis y además calcula las coordenadas de la búsqueda y mostrar un resultado en Google Earth.

A continuación se muestran los diagramas de actividad y de secuencia de una búsqueda sin beneficios.

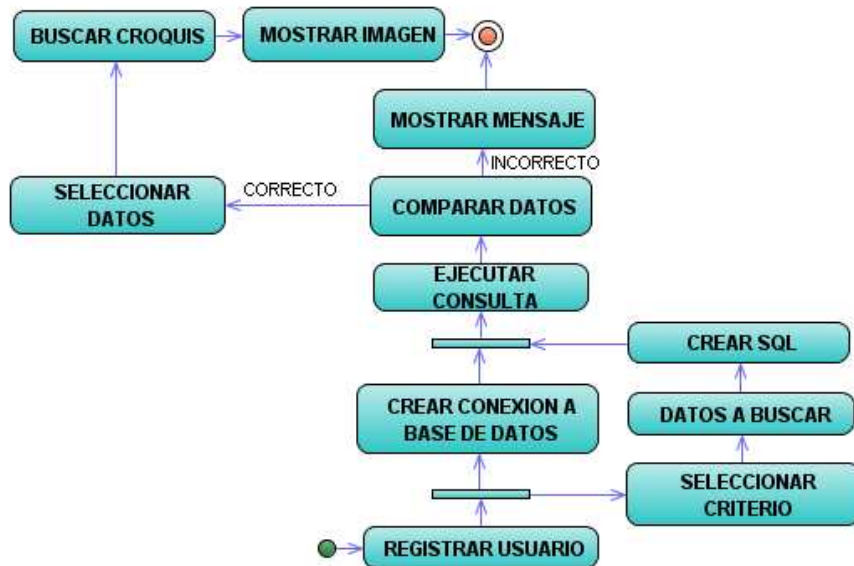


Figura 26. Diagrama de actividad búsqueda sin beneficios

Fuente: El Autor

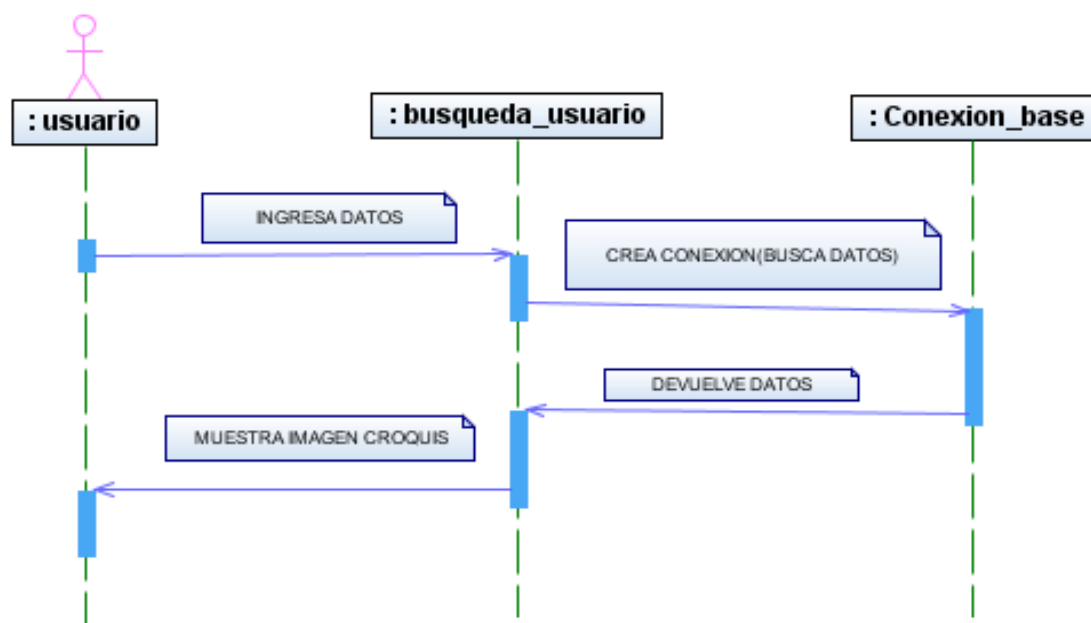


Figura 27. Diagrama de secuencia búsqueda sin beneficios

Fuente: El Autor

La búsqueda con beneficios implica que al momento de culminar la exploración de los datos en el croquis puede el usuario ver una imagen satelital proporcionada por el sistema Google Earth, la cual será especificada por el sistema “Sysgeo” obteniéndole gracias a un cálculo de coordenadas y a la relación de la información que anteriormente se ha realizado.

Los diagramas de actividad y secuencia que representan estas tareas se muestran a continuación.

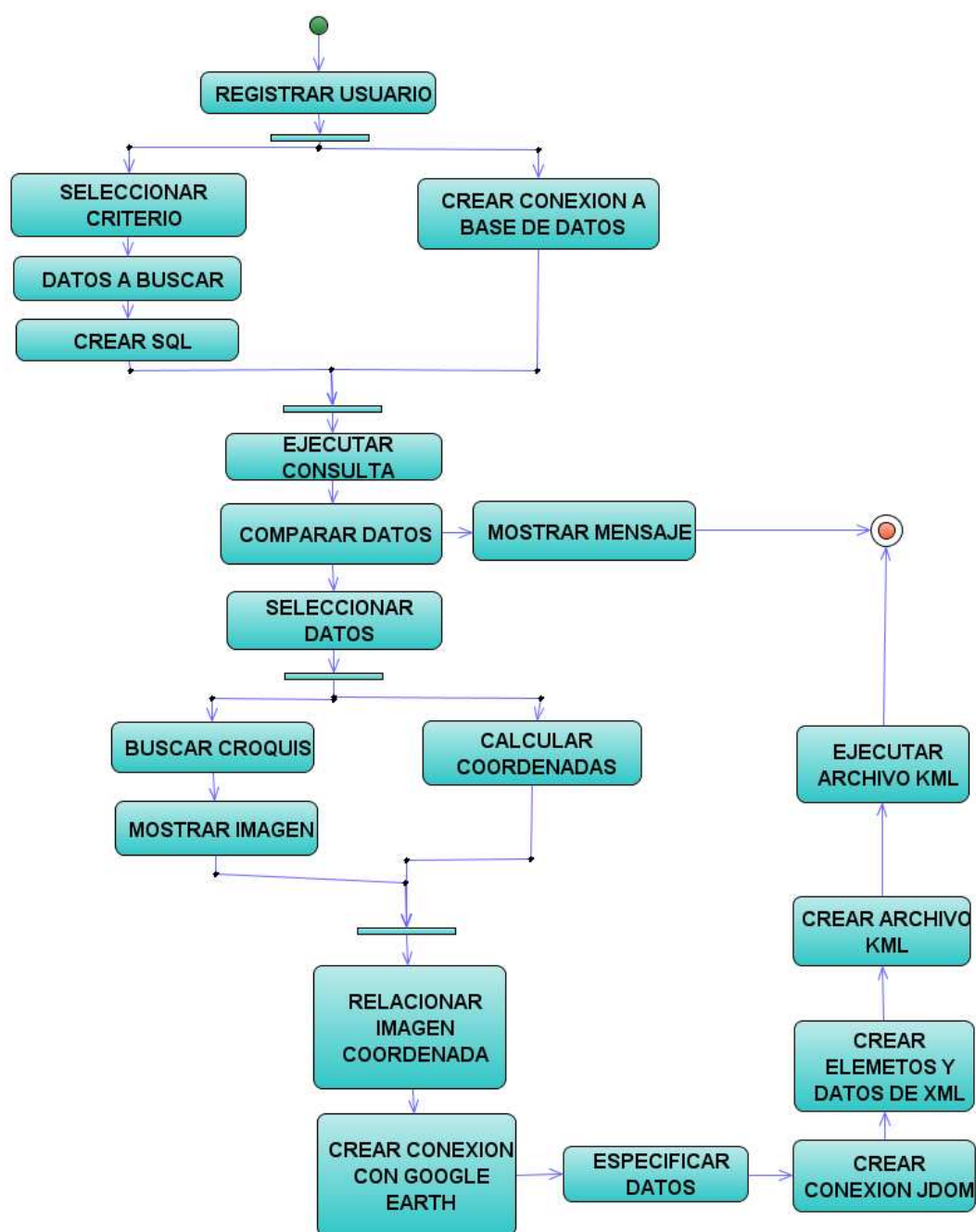


Figura 28. Diagrama de actividad búsqueda con beneficios

Fuente: El Autor

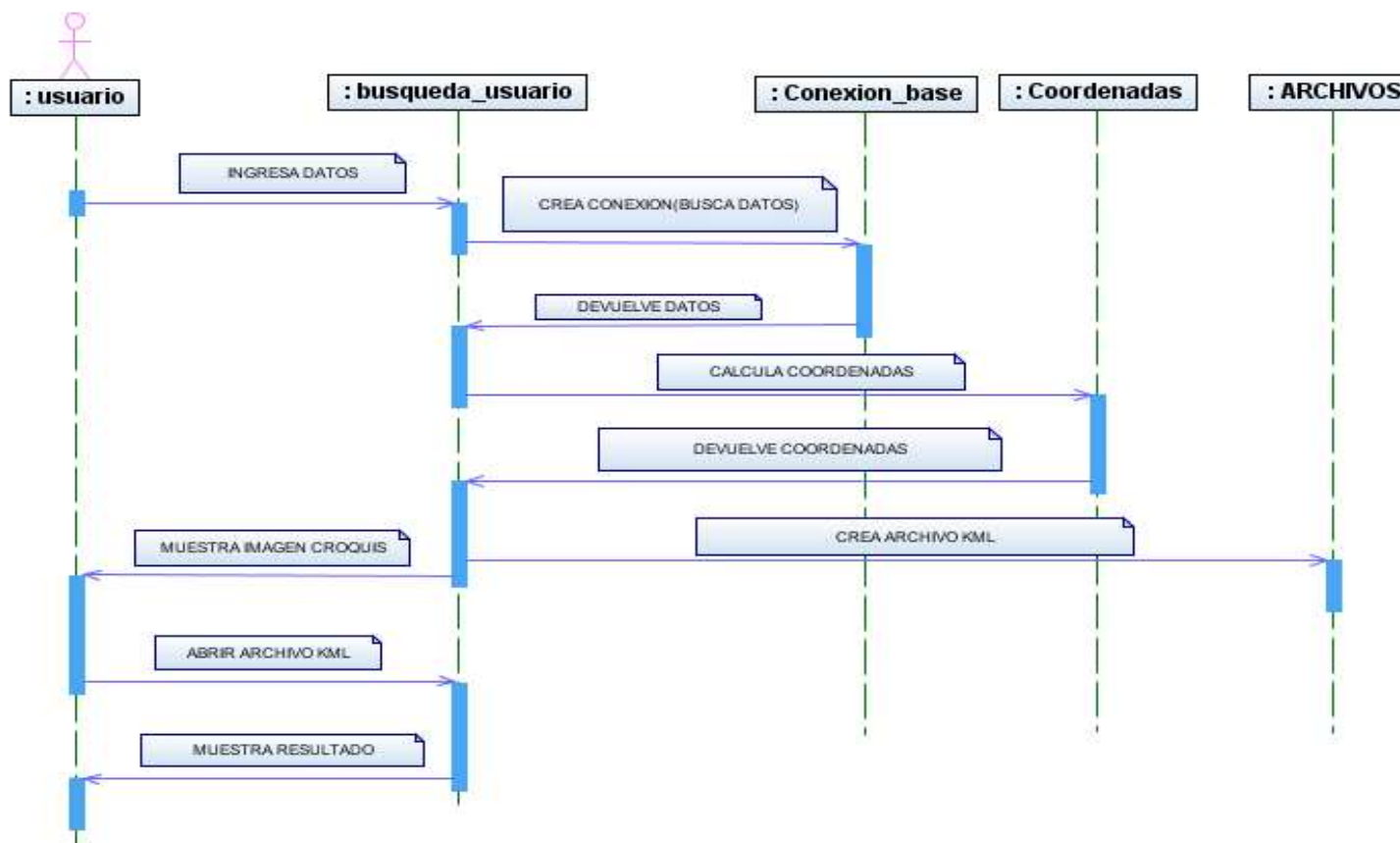


Figura 29. Diagrama de secuencia búsqueda con beneficios

Fuente: El Autor

3.5. EL DISEÑO ARQUITECTÓNICO.

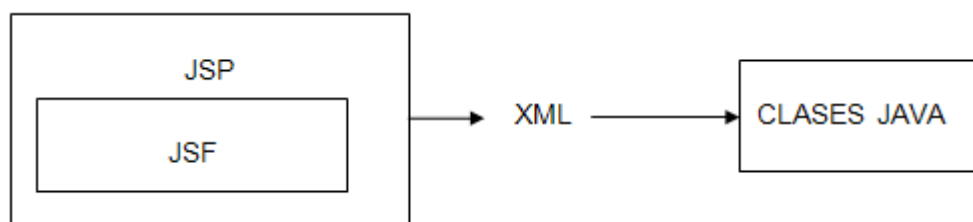


Figura 30. Funcionamiento interno del sistema

Fuente: El Autor

El funcionamiento interno de la interfaz es como se indica en el Figura anterior el código JSF se incrusta dentro del JSP, La comunicación entre la interfaz JSF y las Clases JAVA se realiza a través de un archivo XML

3.6. DISEÑO DEL SISTEMA

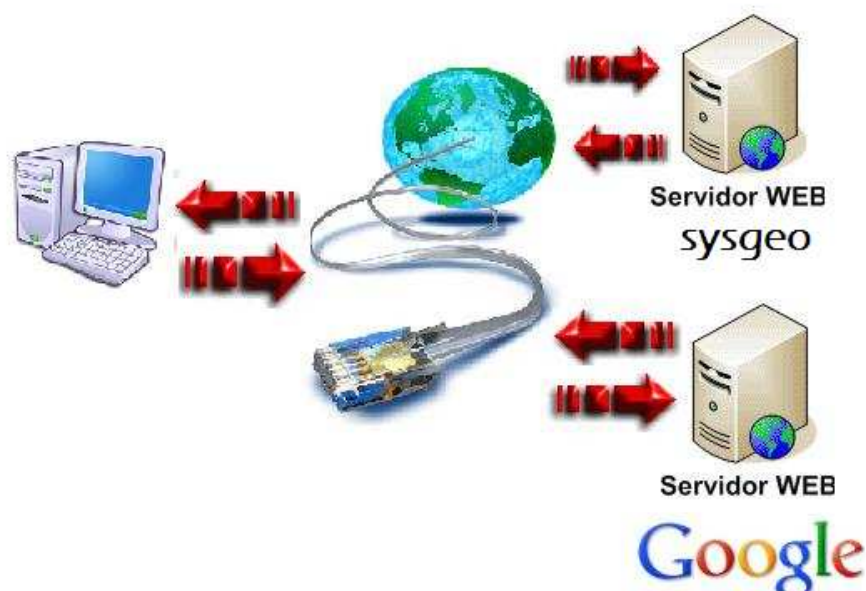


Figura 31. Interfaz de usuario visitante

Fuente: El Autor

El sistema será colocado en un servidor web. El cliente podrá ingresar a través del internet, para de esta manera poder realizar sus consultas, las cuales serán

satisfechas por el aplicativo mostrando una pantalla con el resultado y colocando a disposición del usuario un resultado diferente en el sistema Google Earth.

3.7. EL DISEÑO DE LA INTERFAZ.

3.7.1. INTERFAZ USUARIO VISITANTE

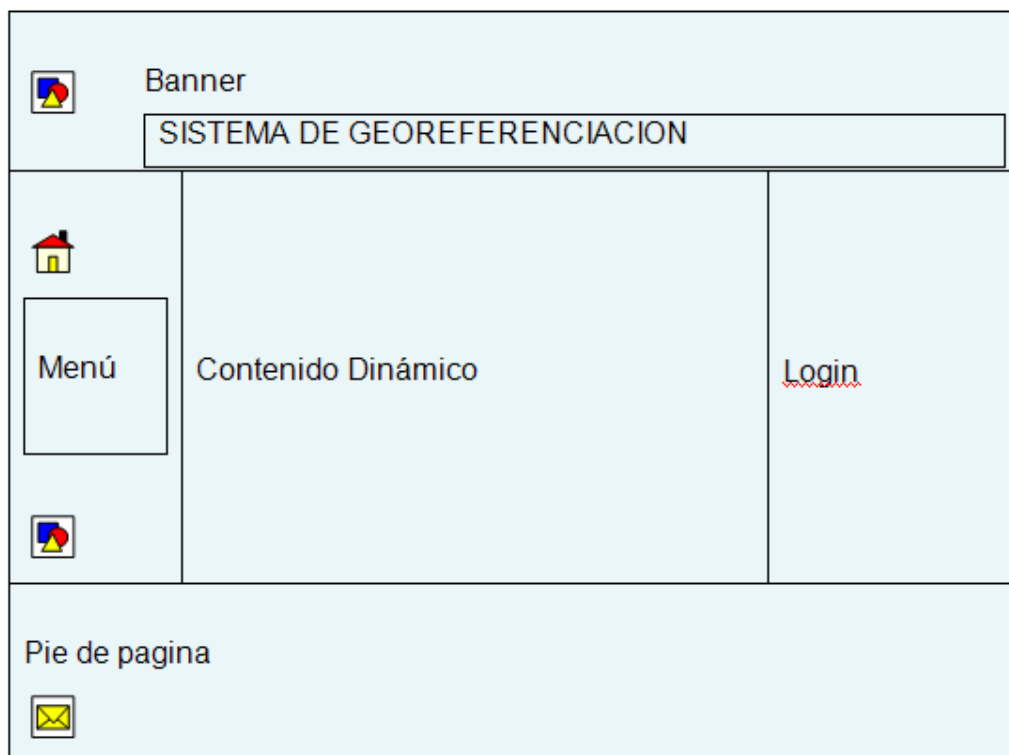


Figura 32. Interfaz de usuario visitante

Fuente: El Autor

El usuario visitante tendrá acceso al aplicativo pero con ciertas restricciones, podrá navegar, dar sugerencias e inclusive realizar búsquedas pero solo interactuara con lo básico de una imagen del croquis.

3.7.2. INTERFAZ DE USUARIOS REGISTRADOS

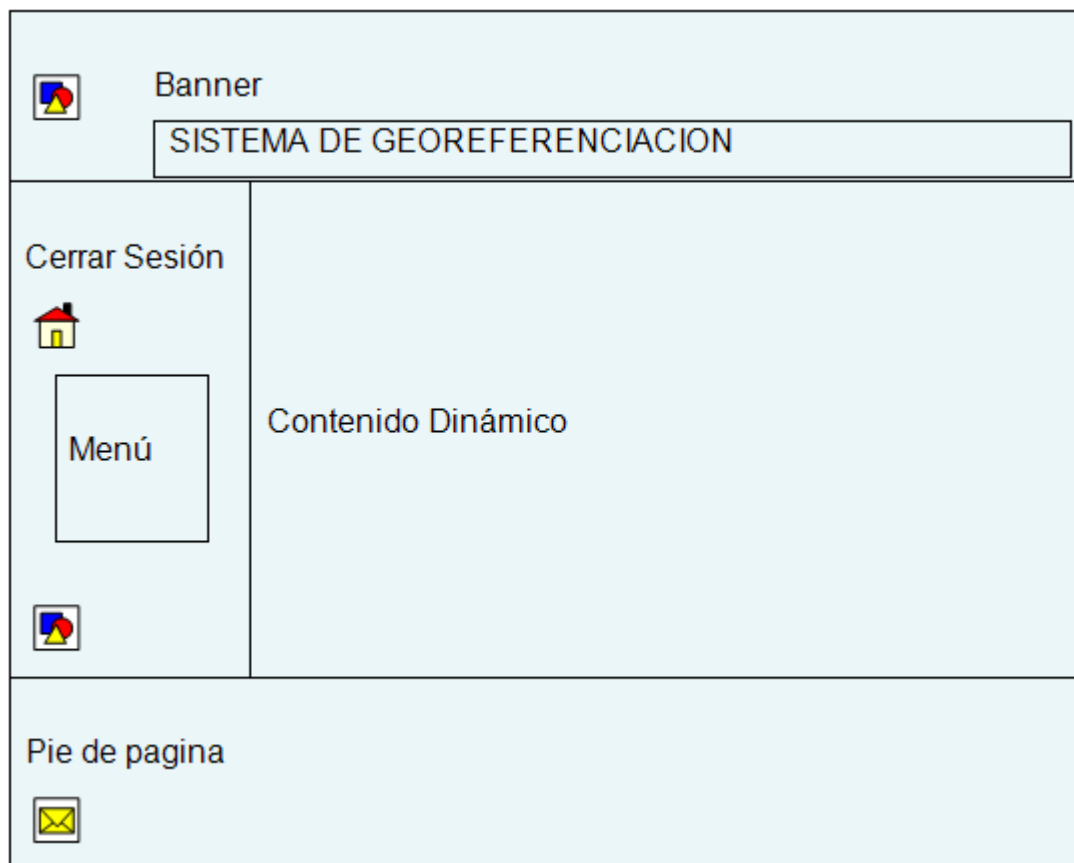


Figura 33. Interfaz de usuario registrado

Fuente: El Autor

El usuario registrado podrá navegar por el aplicativo, realizar búsquedas más especializadas no solo con el croquis de la ciudad sino también tendrá la ventaja de llamar a Google Earth desde la aplicación con referencias del plano de Quito.

3.7.3. INTERFAZ USUARIO ADMINISTRADOR

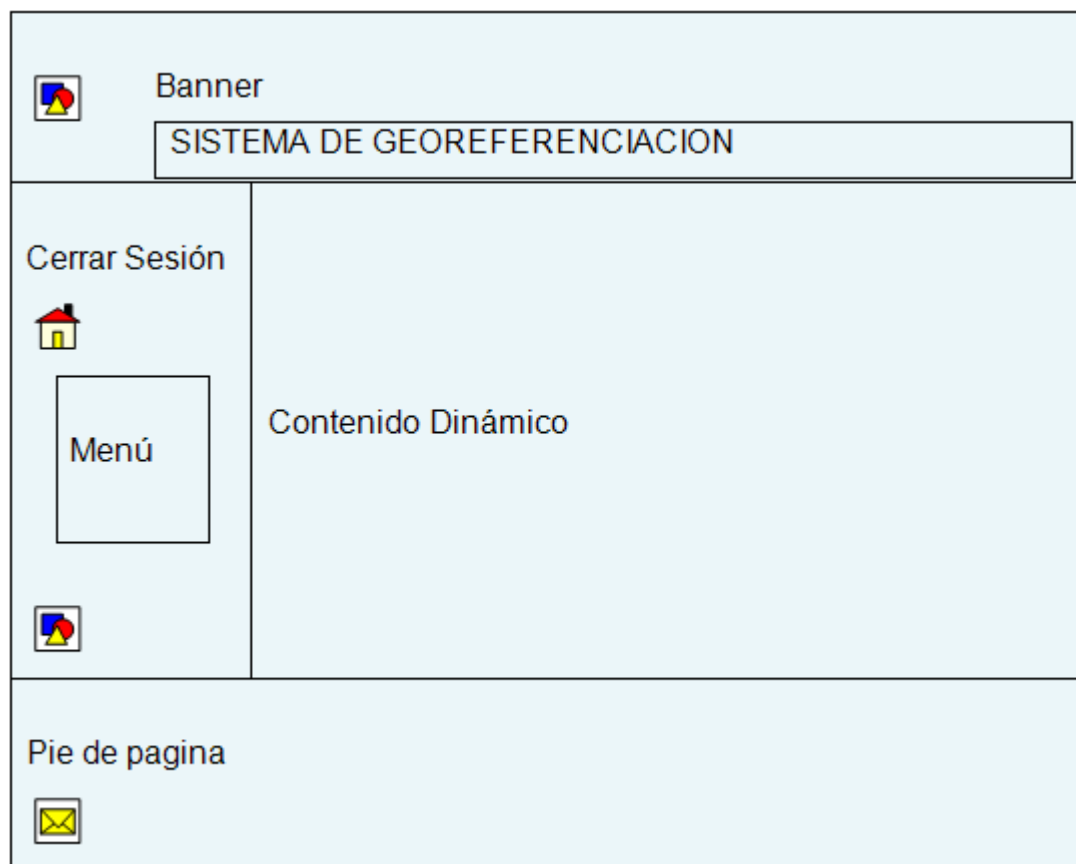


Figura 34. Interfaz de usuario Administrador

Fuente: El Autor

El menú del administrador le guiará hasta las otras interfaces en las que se encuentra el ingreso de datos y asignaciones o relaciones de la información permitiendo de esta manera realizar búsquedas más especializadas, pudiendo modificarlas hasta conseguir un resultado eficiente y exacto. Los menús variarán según el tipo de usuario que se registre, permitiendo el acceso solo a las partes del aplicativo que su registro le permite dando de esta manera seguridad al sistema, para que beneficiarios que no tengan los accesos totales de él no puedan cambiar la información almacenada.

3.8. DISEÑO DE LA SALIDA.

La información que el usuario recibirá es de tipo gráfica ya que la primera respuesta del sistema es una imagen del plano de la ciudad de Quito, luego de la

cual se puede obtener una imagen satelital de la ciudad gracias al sistema Google Earth, Con el cual las personas podrán realizar hasta impresiones de sus búsquedas. El sistema se podrá levantar en un servidor web por lo que todas las personas tendrán acceso a este y revisar su información deseada.

3.9. DISEÑO DE LA BASE DE DATOS

La base de datos está diseñada para poder almacenar datos y relacionarlos de una manera eficaz, para evitar redundancias o ambigüedades y obtener los mejores resultados al momento de empezar las consultas que se lleven adelante.

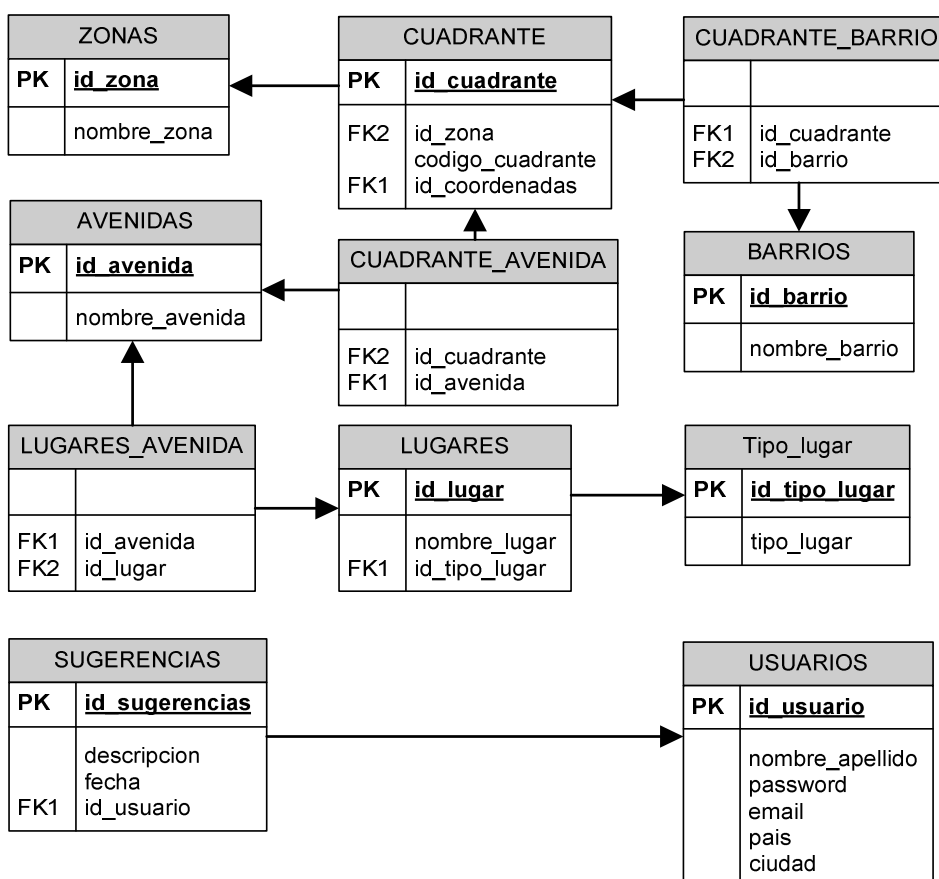


Figura 35. Modelo conceptual de la base de datos

Fuente: El Autor

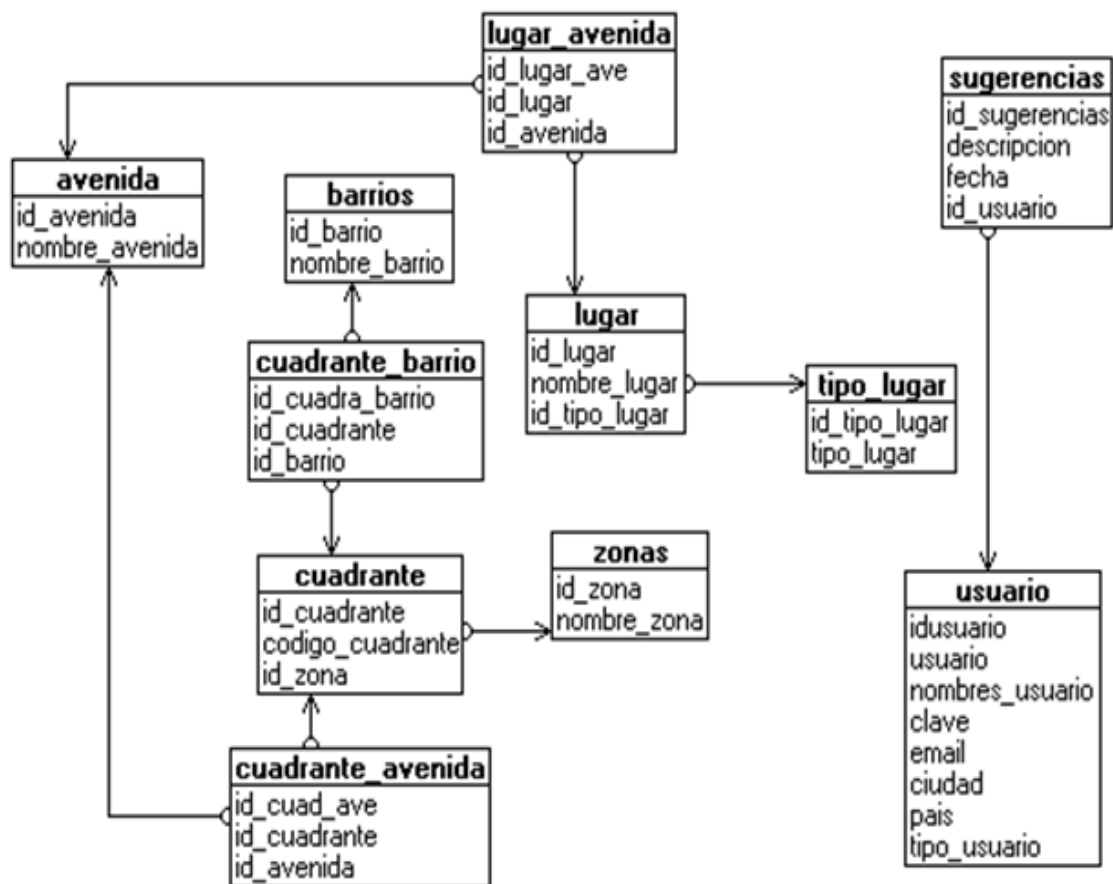


Figura 36. Modelo físico de la base de datos

Fuente: El Autor

Al sistema puede ingresar cualquier usuario con su debida autenticación, mientras esta opción no se realice solo tendrá acceso a las interfaces habilitadas para el visitante, pero cuando un usuario se registra los datos de este pueden ser encriptados por parte del sistema gracias a un código que se ha implementado el cual permite dar seguridad a las claves de acceso de cada cliente.

MAPA DE SITIO DEL SISTEMA DE GEOREFERENCIACION DE LA CIUDAD DE QUITO “SYSGEO”:

3.9.1. USUARIO VISITANTE

En Home el visitante conoce la información de la empresa. Para la Búsqueda General se lo podrá realizar por lugar, por barrio y calle obteniendo una imagen del mapa de Quito de la zona deseada. Sugerencia: el usuario podrá dejar su respectivo comentario acerca del sitio y de los servicios que se le ofrece. Habrá un formulario para el “Regístrate” el mismo que permite la creación de nuevos usuarios. Ingresar al Sitio Web (login): permite mediante un Nick y contraseña ingresar a la página de cliente del sitio y acceder a mejores beneficios

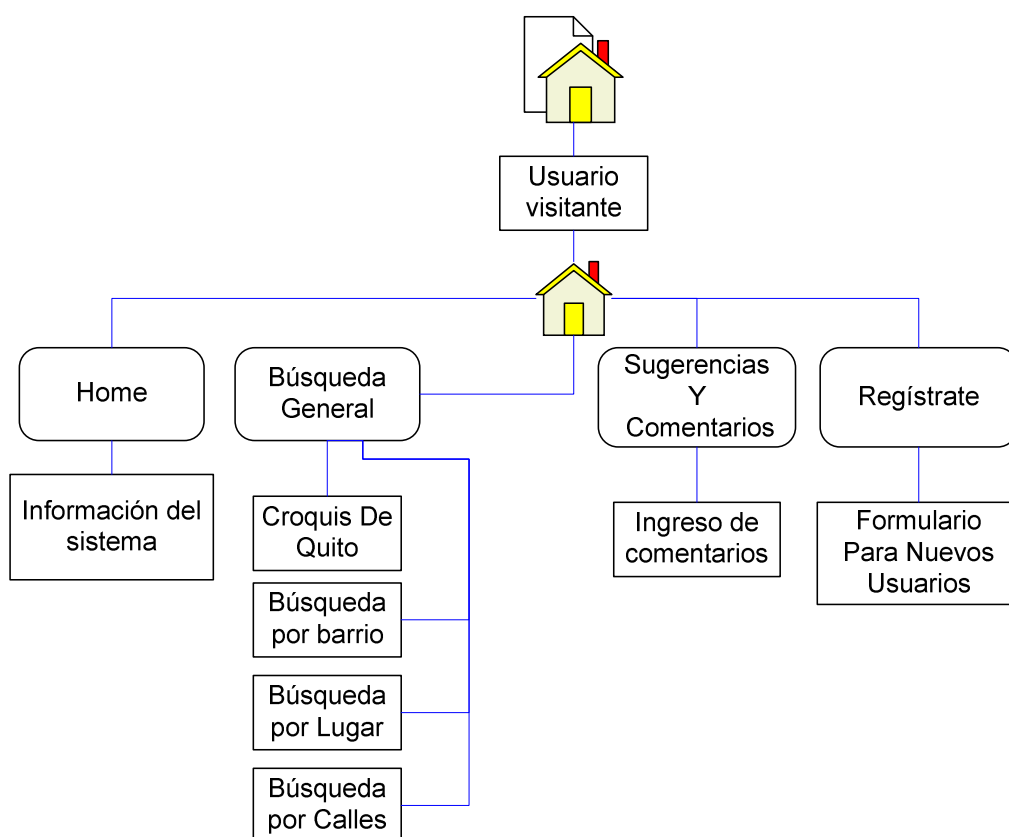


Figura 37. Mapa de sitio usuario visitante

Fuente: El Autor

3.9.2. USUARIO REGISTRADO

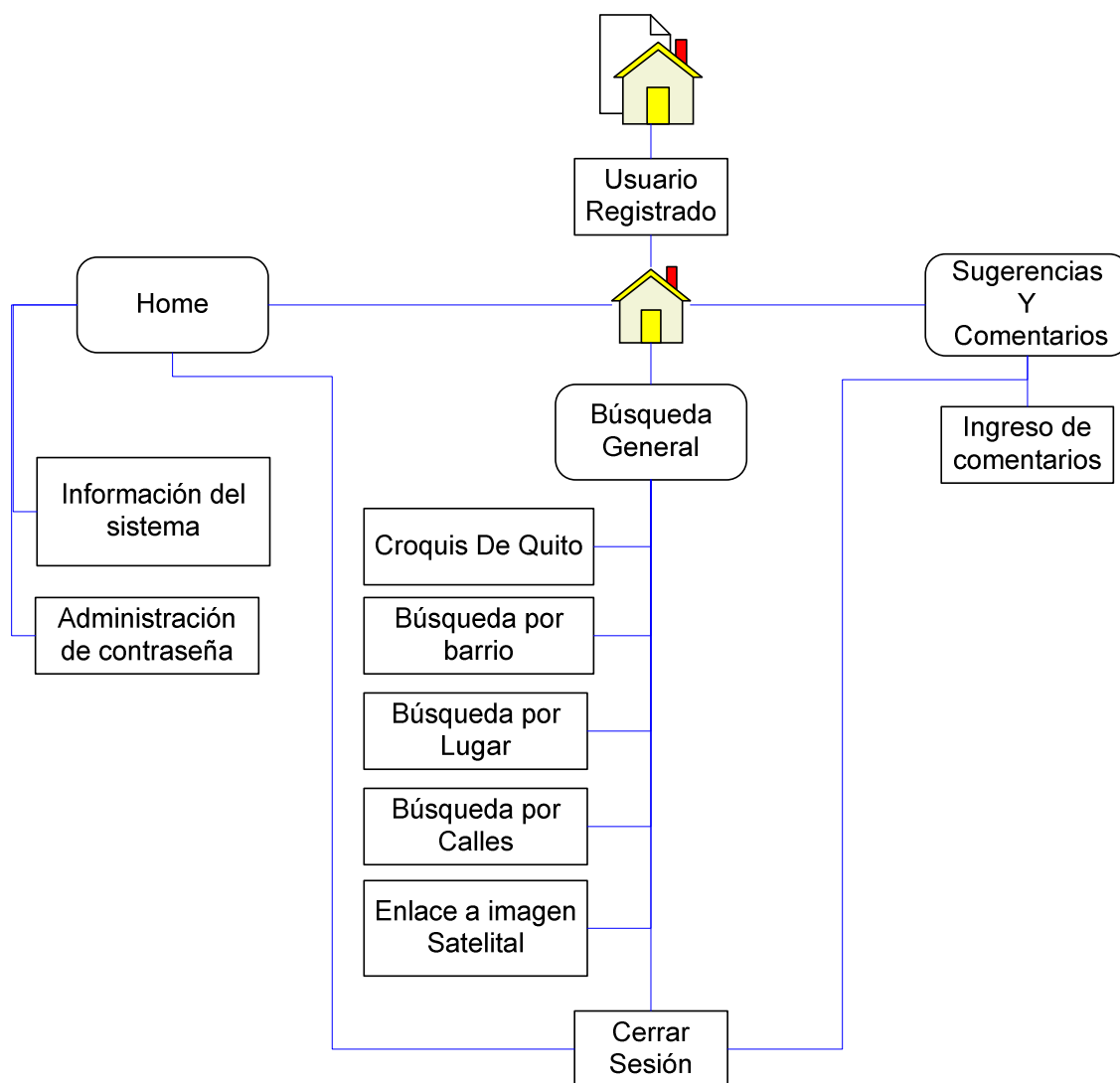


Figura 38. Mapa de sitio usuario registrado

Fuente: El Autor

En el usuario registrado podrá también tener acceso al home y sugerencias del sistema, también podrá realizar búsquedas por lugar, barrio o calle obteniendo una imagen y adicionalmente podrá llamar a Google Earth para mostrar una imagen satelital de la ciudad de Quito.

3.9.3. USUARIO ADMINISTRADOR

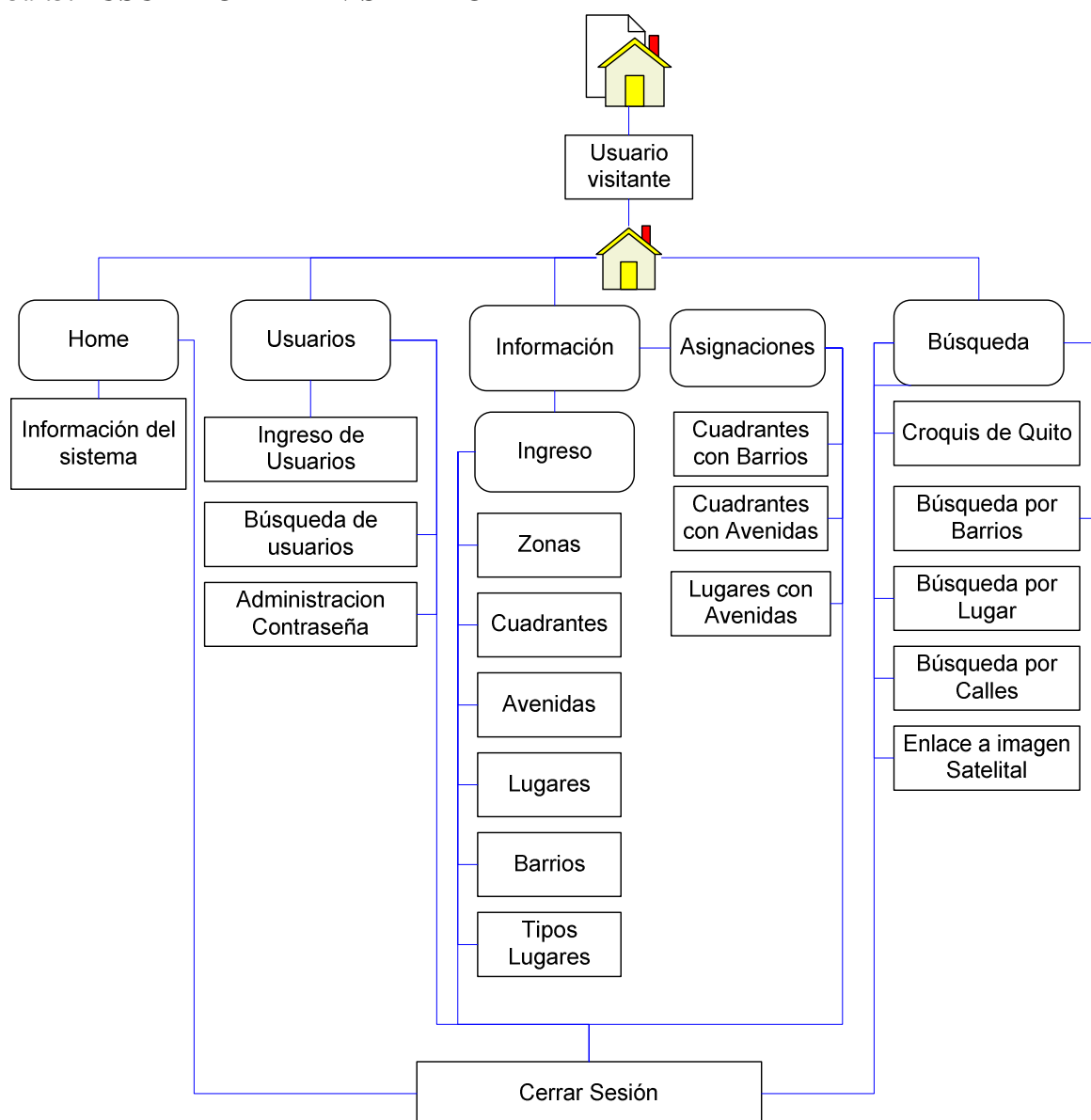


Figura 39. Mapa de sitio usuario registrado

Fuente: El Autor

En el usuario administrador se puede acceder a varias funciones como: la administración de usuarios en el cual puede realizar consultas ya sea por país, ciudad o nombre; el administrador podrá también hacer cambio de contraseña.

Además podrá tener acceso al ingreso de información, asignaciones y consultas de los datos ingresados como avenida, lugar o barrio.

CAPÍTULO IV

CONSTRUCCIÓN

En este capítulo se mostrara los pasos que se siguieron para la construcción del sistema especificando, entorno de desarrollo, tecnologías de programación, servidores de trabajo, además como realizar las diferentes conexiones entre estas.

4.1. ENTORNO DE DESARROLLO INTEGRADO

NETBEANS 6.1

El entorno de desarrollo en el cual se crea el sistema es Netbeans el cual con sus diferentes módulos puede ser capaz de desarrollar de una manera rápida y sencilla cualquier aplicación java de escritorio como de Web.

NetBeans es un IDE (Entorno de desarrollo integrado) de código abierto de gran éxito con una gran cantidad de usuarios. Fue creado por la Sun Microsystems y continúa siendo el patrocinador principal. NetBeans ha tenido una gran aceptación por parte de los desarrolladores Java además este también puede ser utilizado como un desarrollador de otros lenguajes de programación, es un producto libre y gratuito sin restricciones de uso.

La ventaja de principal de utilizar el IDE Netbeans es que fue creado por la misma empresa del lenguaje java por lo cual no tiene ningún inconveniente en acoplarse a nuevas tecnologías de este lenguaje.

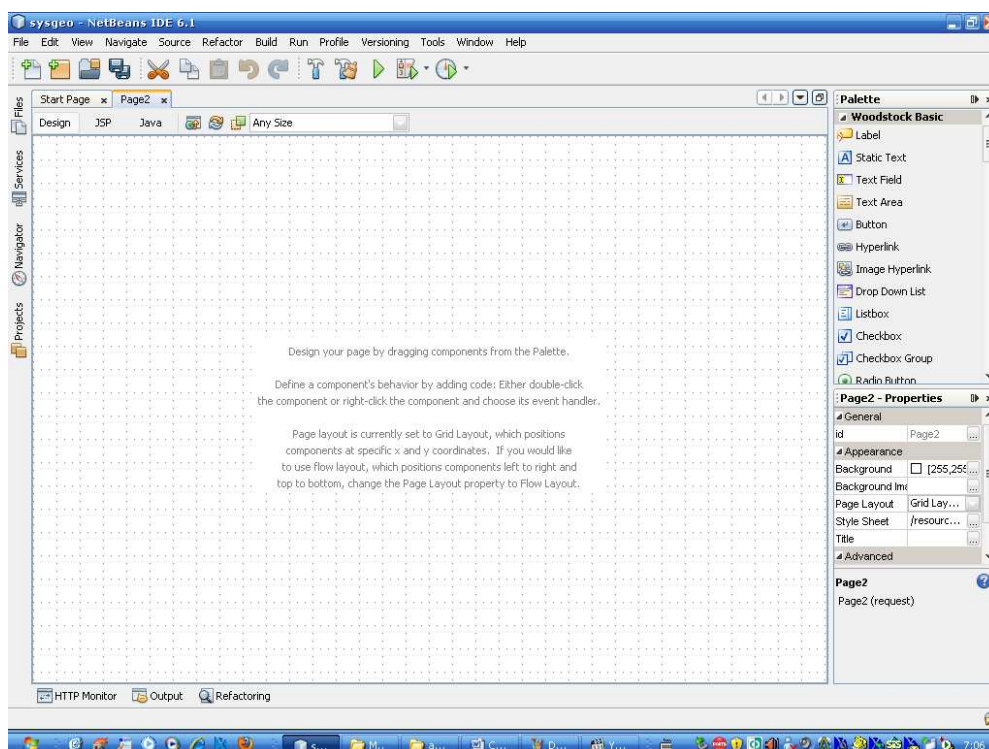


Figura 40. Interfaz Grafica de Netbeans

Fuente: El Autor

4.2. VISUAL WEB JSF

La programación en Visual Web JSF solo es posible en pocos entornos de desarrollo este Plugin esta especificado directamente para funcionar con NetBeans el cual es de gran ayuda, ya que ahorra tiempo y las configuraciones con el XML se realizan de una manera automática dejando solo la lógica de negocio al programador.

La programación visual se realiza de una manera simple y como se puede observar en el Figura N°. 33. Está dividida en tres partes la forma visual, la programación JSF que se incrusta dentro del JSP y la parte de java. La conexión entre estas se realiza gracias al Plugin(Visual JSF).

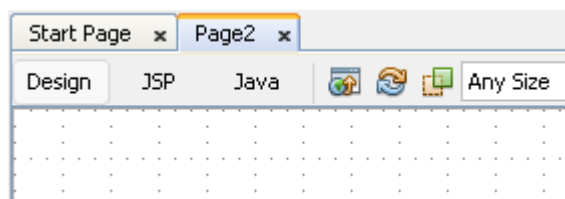


Figura 41. Visual Web JSF

Fuente: El Autor

Los objetos que son agregados a la interfaz no se pueden utilizar directamente desde la clase java hasta que se realice un Add Binding Attributes como se muestra en el Figura N°. 34. El cual es necesario para la utilización de este objeto en programación.

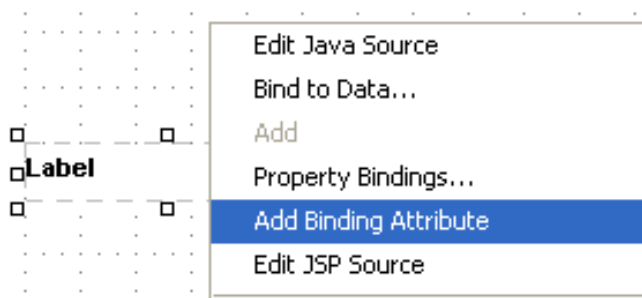


Figura 42. Add Binding Attribute

Fuente: El Autor

La parte restante de la programación es igual al desarrollar interfaces para escritorio.

4.3. FRAGMENTOS

Los fragmentos son archivos que pueden ser añadidos a cualquier interfaz facilitando el diseño al programador, esto ayuda a que no se repitan líneas de código de manera innecesarias.

Los fragmentos al igual que las interfaces están asociados con una clase java, los pasos para crearlos son:

- Utilizar la Herramienta Page Fragment Box que se encuentra en el cuadro de herramientas

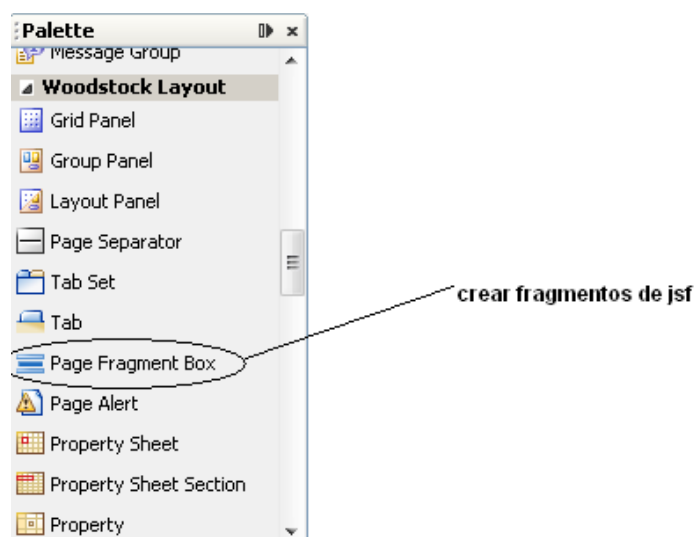


Figura 43. Page Fragment Box

Fuente: El Autor

- Al dibujar este componente en la interfaz aparecerá un cuadro de dialogo en el cual se debe colocar el nombre de un Fragmento o en su lugar uno nuevo que se desee crear.

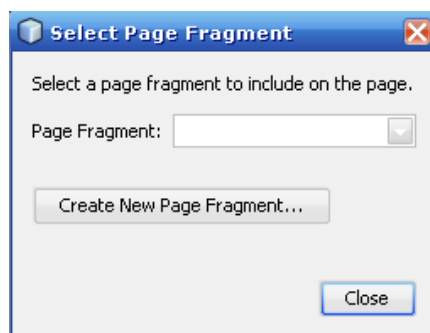


Figura 44. Selección de Fragmentos Fuente: El Autor

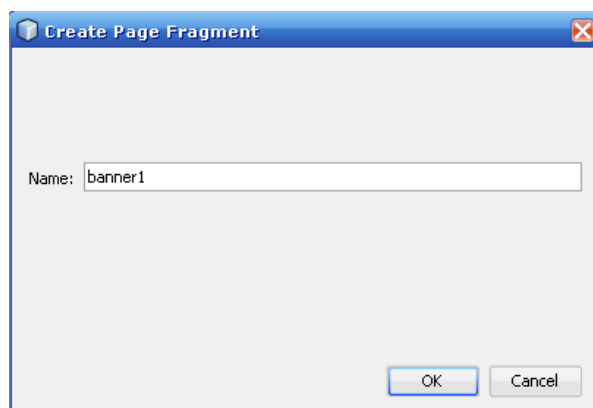


Figura 45. Fragmento nuevo

Fuente: El Autor

Luego de lo cual se debe realizar el diseño de fragmento colocando imágenes u objetos. El resultado final será como se indica en el gráfico N°. 39. En el cual se muestra como resultado un banner.

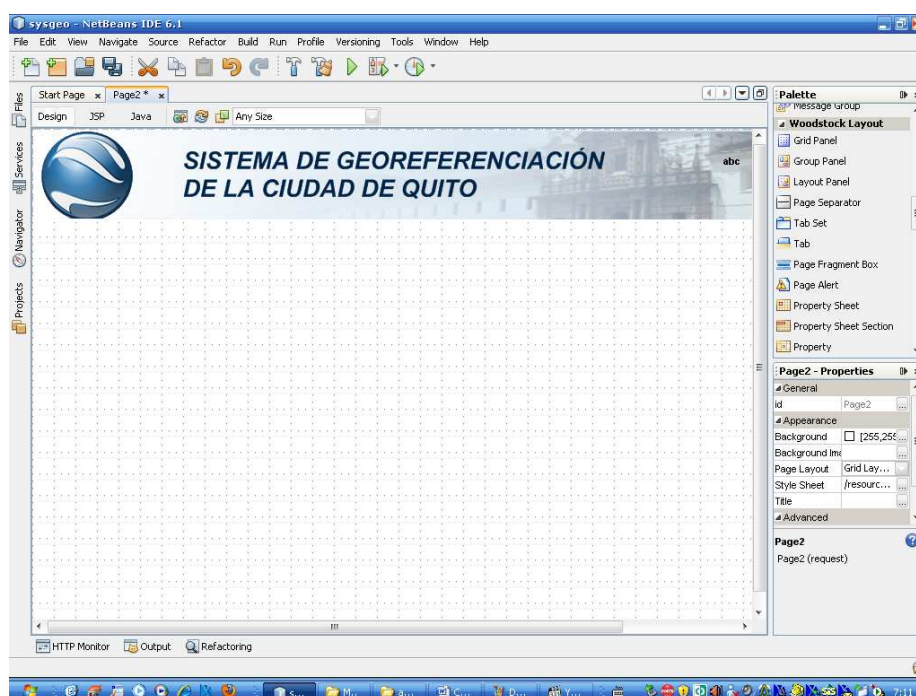


Figura 46. Selección de Fragmentos

Fuente: El Autor

Además estos pasos se realizaron para crear los banners, pie de páginas, Autenticación y menús.

4.4. SESIONES

En el caso de Geosys se consta con tres tipos de usuarios entre los que se tiene Invitado, registrado y administrador.

Una de las facilidades que ofrece NetBeans es poder crear navegaciones dinámicas solo con relacionar las interfaces y colocar un nombre con el cual se tomara la decisión de que interfaz se desplegara a continuación. En este caso **case1** para usuario registrado y **case2** para usuario administrador como lo indica el Figura N°. 40.

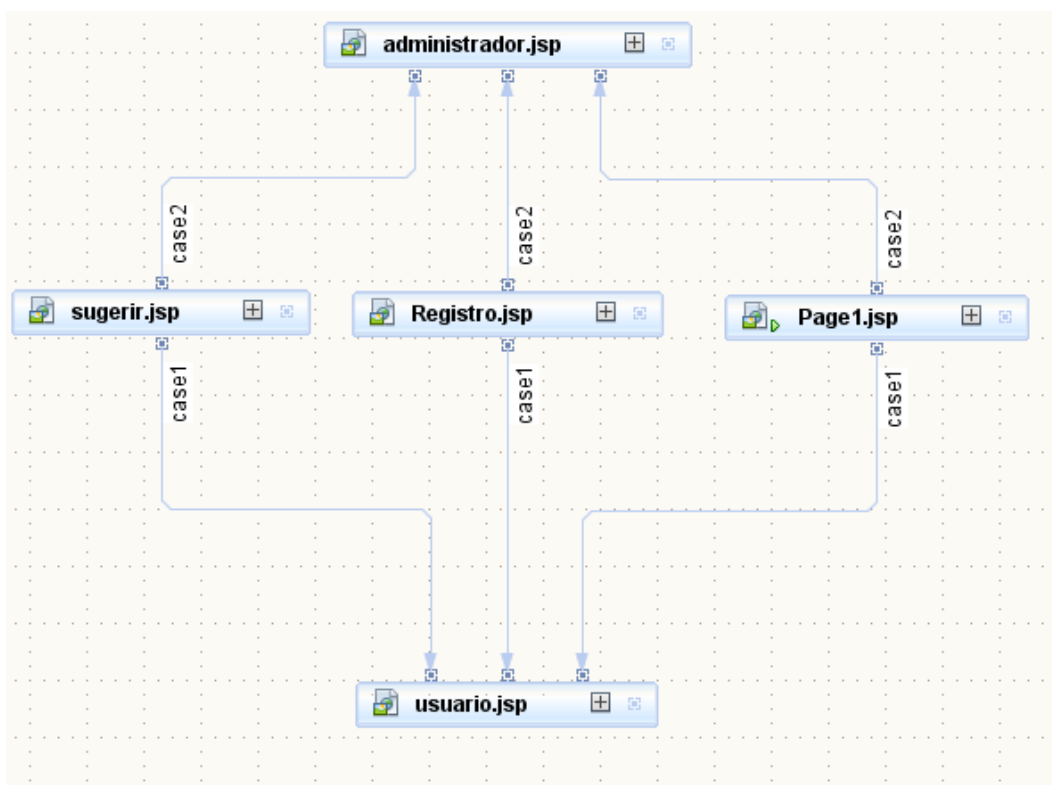


Figura 47. Relaciones entre interfaces

Fuente: El Autor

Para poder hacer uso de las relaciones anteriores se debe colocar unas líneas de código especificando el valor a devolver. Esto se debe colocar por ejemplo en un botón indicando el tipo de usuario y el retorno que debe tener para cada uno de ellos.

```
if(administrador){
    return "case2";
}else{
    return "case1";
}
```

Figura 48. Relaciones entre interfaces

Fuente: El Autor

Además también se debe crear la sesión para el usuario, indicando los atributos que debe tener, como nombre, tipo y código. Para esto se utiliza la sentencia HttpSession lo cual se expresa de mejor manera en el Figura N°. 42.

```
HttpSession s=(HttpSession)FacesContext.getCurrentInstance().getExternalContext().getSession(true);
s.setAttribute("userName", usuario);
s.setAttribute("hayUsuario", "si");
s.setAttribute("Administrador", "si");
s.setAttribute("codigo", user);
```

Figura 49. Creación de sesiones

Fuente: El Autor

En las interfaces a las que acceden los usuario registrado se agregan las siguientes líneas de código que permitan visualizar cuales son los usuario y que características tiene cada uno.

```
public String getSaludo(){
    HttpSession s=(HttpSession) FacesContext.getCurrentInstance().getExternalContext().getSession(true);
    String hayLog =(String)s.getAttribute("hayUsuario");
    String Usuario="Invitado";
    if(hayLog != null && hayLog.compareTo("si")==0){
        Usuario=(String)s.getAttribute("userName");
        return "Bienvenido: " + Usuario;
    }else{
        return "Bienvenido: INVITADO";
    }
}
```

Figura 50. Búsqueda de usuarios registrados

Fuente: El Autor

Una de las ventajas proporcionadas por el anterior método y tomando en cuenta las características propias de JSF se puede mostrar el nombre del usuario en cada interfaz que se desee, por ejemplo en un Label solo realizando una especificación en el contenido (texto) de este componente que se encuentra en las propiedades.

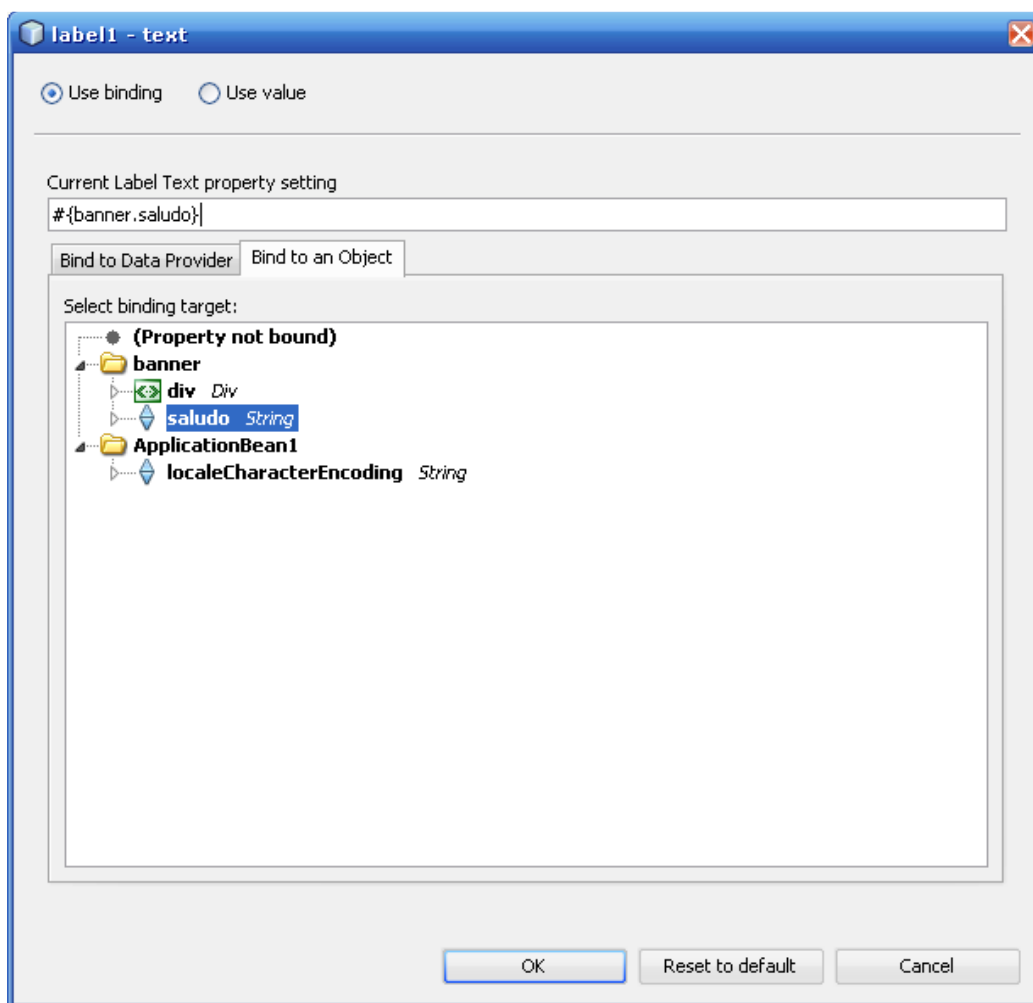


Figura 51. Búsqueda de usuarios registrados

Fuente: El Autor

4.5. JDOM (CREACIÓN DE ARCHIVOS KML)

Los Archivos KML son la conexión con el sistema Google Earth. Los archivos Kml están compuestos de igual manera que un archivo XML por elementos, y uno de las formas de poder trabajar con estos archivos desde java es con JDOM.

JDOM es una librería (jdom.jar) que se puede descargar del internet ya que esta no viene incluida en la maquina virtual del java y su modo de trabajar es muy simple. Teniendo en cuenta que los archivos tienen elementos y algunos subelementos.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml>
  <PhotoOverlay>
    <name>K25</name>
    <Camera>
      <longitude>-78.484552681</longitude>
      <latitude>-0.149193543000000004</latitude>
      <altitude>1056.009012417283</altitude>
      <heading>-2.80932092409122e-007</heading>
      <tilt>0</tilt>
      <roll>0</roll>
      <altitudeMode>relativeToGround</altitudeMode>
    </Camera>
    <Icon>
      <href>K25.jpg</href>
    </Icon>
    <ViewVolume>
      <leftFov>-25.055</leftFov>
      <rightFov>25.055</rightFov>
      <bottomFov>-25.055</bottomFov>
      <topFov>25.055</topFov>
      <near>369.603</near>
    </ViewVolume>
    <Point>
      <altitudeMode>relativeToGround</altitudeMode>
      <coordinates>-78.484552681,-0.149193543000000004,1056.009012417283</coordinates>
    </Point>
  </PhotoOverlay>
</kml>
```

Figura 52. Archivo KML

Fuente: El Autor

```
public Document genXML(String alias, String longitud, String latitud) {
    Document doc = new Document();
    Element root = new Element("kml");
    Element PhotoElem = new Element("PhotoOverlay");
    Element name1=new Element("name");
    name1.setText(alias);
    PhotoElem.addContent(name1);
    Element camera=new Element("Camera");
    PhotoElem.addContent(camera);
    Element longitude=new Element("longitude");
    longitude.setText(longitud);
```

Figura 53. Diseño del KML

Fuente: El Autor

La mayoría de los archivos KML tienen la misma estructura solo varían las coordenadas y los nombres de los objetos por lo cual esto es un parámetro que controla el sistema, para crear el archivo se utiliza las siguientes líneas de código.

```
public void dumpXML(Document doc, String filePath) throws IOException {
    XMLOutputter outputter = new XMLOutputter();
    FileWriter writer = new FileWriter(filePath);
    outputter.output(doc, writer);
}
```

Figura 54. Creación del archivo KML

Fuente: El Autor

4.6. ARCHIVOS WAR

Estos archivos son indispensables ya que con su creación nos podemos olvidar de cualquier configuración en el servidor este archivo contiene todas las configuraciones, enlaces, librerías y las clases de trabajo. Existen dos formas de crear el archivo war una es desde el símbolo del sistema y la otra es desde el mismo IDE, en Netbeans su creación se realiza de la siguiente manera:

- Se selecciona el proyecto
- Menú Build
- Se elige la opción Clean and Build Main Project,
- El archivo War se crea en el directorio Dist de nuestro proyecto Java

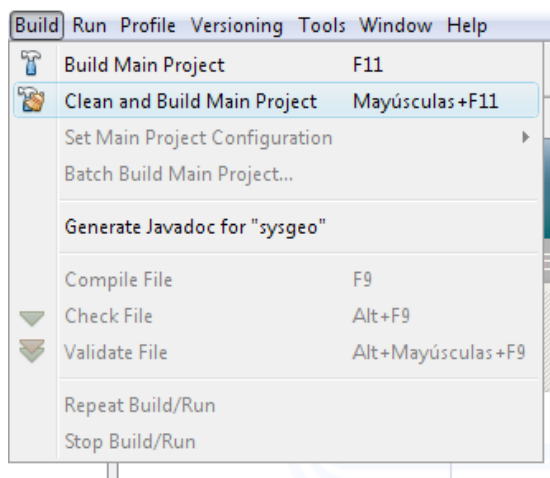


Figura 55. Creación del archivo War

Fuente: El Autor

Este Archivo War se ubica en el servidor WEB. Aun cuando este funciona por defecto en el servidor Glassfish el cual fue creado por la Sun Microsystem como software libre.

4.7. CONEXIÓN BASE DE DATOS

La base de datos escogida para trabajar con el proyecto es MYSQL para la cual se necesita un conector (mysql-connector-java-5.0.8-bin.jar) el cual se puede descargar de la misma página de MYSQL

La conexión a la base de datos se realiza de la misma manera que todas, solo se debe tener en cuenta el driver y el url de la base de datos.

```
private String controlador="com.mysql.jdbc.Driver";
private String conec="jdbc:mysql://localhost/sysgeo";
private String user="root";
private String clave="sys";
private Connection conexion;
private Statement st;
private ResultSet rs;
```

Figura 56. Datos conexión a la base de datos

Fuente: El Autor

Para ejecutar la sentencia se realiza de una manera simple especificando el controlador, url, usuario y password de la base de datos. Además de la debida sentencia sql.

```
public void Guardar(String sql1){
    try{
        Class.forName(controlador);
        conexion= DriverManager.getConnection(conec, user, clave);
        st= (Statement) conexion.createStatement();
        st.execute(sql1);
    }catch(Exception e){
    }
}
```

Figura 57. Datos conexión a la base de datos

Fuente: El Autor

REQUISITOS PARA EL FUNCIONAMIENTO DE JSF

JSF necesita de ciertas características para su funcionamiento correcto, los requisitos que se deben tener son los siguientes.

- Versiones de Java.
 - JDK 1.4.x
 - JDK 1.5.x
- Contenedor de Servlet.
 - Tomcat
 - JBoss
 - Glassfish

Cualquier motor que cumpla con la especificación 2.3 debería valer, pero con la versión jsp-1.2.jar no es suficiente por lo que se debe agregar una jar adicional que viene con la distribución la jsp-2.0.jar

Además hay que tener en cuentas las siguientes consideraciones:

- El motor de servlet debe ser compatible con la especificación 2.3 y las JSP deben ser acordes a la especificación 1.2.
- JSF únicamente soporta peticiones realizadas con POST.
- La especificación no obliga a que haya validaciones en el cliente, MyFaces y Shale proporcionan esta posibilidad.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Con la creación del sistema de georeferenciación de la ciudad de Quito (Sysgeo) se obtuvo una nueva perspectiva del funcionamiento de los sistemas informáticos aplicados hacia la ayuda a la colectividad.
- Al crear el sistema se pudo conocer muchas de las variantes que ha tenido la ciudad y reformas a su estructura arquitectónica y crecimiento en los últimos años, en parte debido al aumento de la población en la ciudad. Los cambios efectuados en la ciudad son muy notorios al comparar los diferentes planos existentes así como las imágenes satelitales, que nos muestran un vistazo hacia el pasado.
- Google earth tiene una gran variedad de ediciones para cada tipo de usuario, y dependiendo de esta Ud. podrá obtener mejores resultados, aunque en la mayoría de las versiones tiene el inconveniente de que las imágenes satelitales están desactualizadas.
- Java es un lenguaje al que se le puede extender características con solo instalar nuevas librerías, estas librerías solo se deben copiar a la carpeta de librerías JAVA pudiendo de esta manera utilizar opciones extras a lo que originalmente el lenguaje tenía.

- Los componentes utilizados en el desarrollo del sistema se pueden describir de una manera distinta a la del funcionamiento de los utilizados en otras tecnologías facilitando el trabajo obteniendo una verdadera programación por capas.
- La programación en JSF es más simple de lo que se esperaba además con la existencia de muchos entornos de desarrollo que incluyen esta nueva forma de crear páginas web se pueden obtener mejores resultados que con otras tecnologías.
- JSF no está sujeto a las herramientas que posee HTML por lo que se pueden crear mejores interfaces de usuario con nuevos elementos implementando mejores características a nuestras aplicaciones en desarrollo web.
- La información utilizada para el desarrollo de esta aplicación es muy accesible ya que se puede conseguir una guía actualizada de la ciudad con los datos necesarios y detallados de toda la ciudad de Quito
- El sistema “sysgeo” al trabajar mediante via web se puede acceder desde cualquier plataforma pero con la limitante del Google Earth ya que no funciona en todos los sistemas operativos.
- Los manuales JSF son casi nulos ya que la mayor parte de la información es de cómo trabaja y cuáles son sus ventajas pero ejemplos físicos son relativamente pocos por lo que si el usuario necesita utilizar herramientas nuevas tendrá que crear clases y enlazarlas a esos objetos.
- Para las operaciones básicas existen plugins que se ocupan de todo lo que se refiere a comunicación dejando solo la programación al desarrollador, pero cuando se utilizan herramientas avanzadas existen problemas de comunicación.

- La parte de la lógica de negocio se realiza completamente solo en java la comunicación entre esta y la capa interfaz se realiza a través de un XML. Por lo que la comunicación entre este es un verdadero campo de acción.

5.2. RECOMENDACIONES

- Para el desarrollo de aplicaciones web se debe comprender que hay incontables herramientas que nos podrían ayudar a su desarrollo, con lo cual los programadores podrían crear e innovar con nuevas características, además de la existencia de librerías que aumentan el funcionamiento y rendimiento del aplicativo.
- Existen muchas tecnologías de creación de sitios web, escoger la adecuada dependerá más de la naturaleza del proyecto y a quien va dirigido. Entre las que se encuentra JSF como una buena tecnología de desarrollo web, que más aprovecha la programación web.
- Antes de dar pasos en la tecnología JSF se debe tener muy buenas bases en java, conocer sus ventajas y desventajas además de los requisitos que este debe incluir para el funcionamiento correcto.
- Al momento de desarrollar aplicaciones web se debe tener en cuenta que lenguaje y servidores se utilizaran ya que la mayoría de ellos se deben agregar atributos y características extras para su funcionamiento.

BIBLIOGRAFÍA

- CEBALLOS SIERRA, Francisco Javier; *Interfaces Graficas y aplicaciones para internet*. 3 edición (2008) , Editorial Ra-Ma
- LEWIS John; CHASE Joseph; *Estructura De Datos Con Java*; Prentice; 1 edición (02/2006)
- CEBALLOS SIERRA, Francisco Javier; *java 2: lenguaje y aplicaciones*: Editorial Ra-Ma; (2006)
- PERRY, Bruce W: *Java Servlet And Jsp Cookbook*; O'Reilly Vlg. GmbH & Co. 1 edición
- MONSON-HAEFEL, Richard;*J2EE Web Services. The Ultimate Guide*; Addison Wesley;1 edition
- TONG, KENT Ka lok *Beginning Jsf 2 Apis And Jboss Seam*; APRESS ; (05/2009)
- JACOBI, Jonas; FALLOWS, John R; *Pro JSF and Ajax. Building Rich Internet Components*; Springer-Verlag GmbH; 1 edición (2006)
- <http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/JSF>
- <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=IntroduccionJSFJava>

ANEXOS

MANUAL DE USUARIO

Sistema de Georeferenciación de la ciudad
de Quito

1. SISTEMA DE GEOREFERENCIACION DE LA CIUDAD DE QUITO

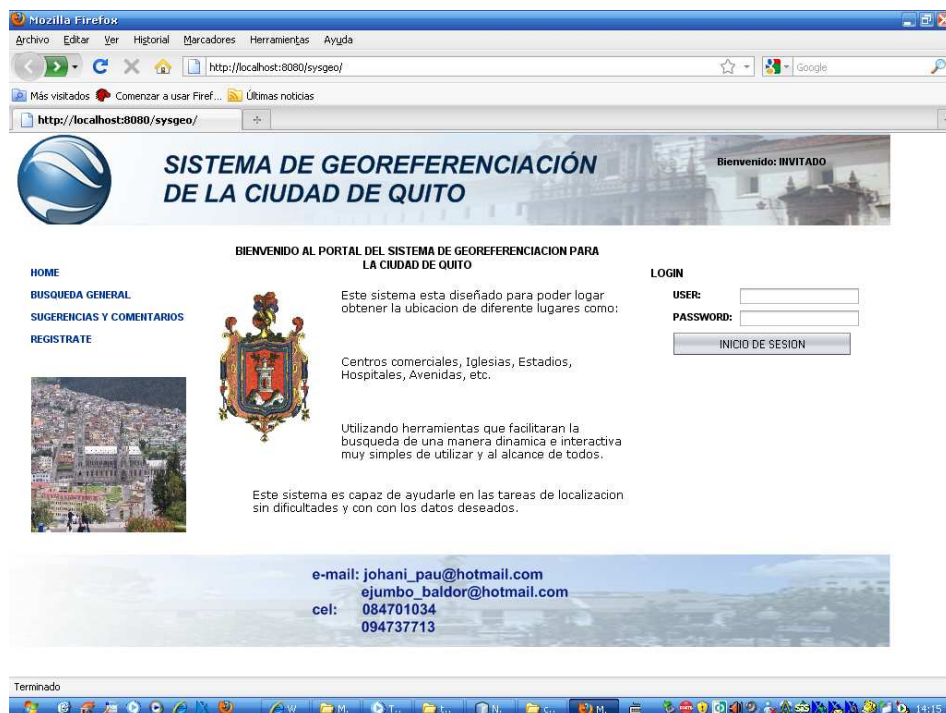


Figura 58. Home

El sistema de georeferenciación de la ciudad de Quito es un aplicativo el cual permite localizar un lugar, una calle, o un barrio de una manera simple e interactiva. El sistema permite que cualquier usuario pueda ingresar y realizar sus búsquedas respectivas en la ciudad. Dando dos formas de mostrar el resultado de la búsqueda. El sistema está diseñado y construido para funcionar en el internet con la tecnología JSF (Java server Faces) conectado a Google Earth el cual servirá como base de datos grafica del sistema.

2. REQUERIMIENTOS DE USUARIO

El aplicativo SYSGEO es una herramienta que funciona en el internet para lo cual el usuario debe de tener la dirección de la misma, además de las siguientes características que debe poseer su computador.

- Conexión a internet mayor a 64 kbps.
- Navegador de internet.
- Tener instalado el sistema Google Earth.
- Resolución de pantalla de 1024 x 700
- Sistema operativo compatible con Google earth.

Los cuales son básicos y se pueden cumplir con facilidad y cualquier usuario puede emplear.

3. AUTENTICACIÓN DE USUARIOS

Una de las primeras tareas que se deben realizar un usuario en el sistema es la autenticación con la cual el podrá iniciar una sesión con los permisos que esta cuenta le permita, además dependiendo de estos autorizaciones podrá obtener resultados del sistema.

LOGIN

USER:

PASSWORD:

Figura 59. Autenticación de usuarios

El sistema acepta tres tipos de usuarios:

- Usuario visitante.
- Usuario registrado.
- Usuario Administrador.

En el propio sistema se pueden registrar los usuarios, que podrán manipular la información que se encuentra almacenada así como modificarla (Usuario administrador).



Figura 60. Mensaje de Bienvenida usuario invitado

Cuando un usuario no se ha registrado le aparecerá un mensaje de **Bienvenido: INVITADO** y solo podrá realizar las actividades básicas permitidas para este tipo de usuario. Este mensaje se mostrara mientras no se realice la autenticación en el sistema.



Figura 61. Mensaje de Bienvenida usuario registrado

Cuando la autenticación ha sido satisfactoria aparecerá el nombre del usuario registrado, dándole una bienvenida y mostrando las paginas que están a su disposición con sus debidos permisos.

Los pasos para realizar esta tarea son simples, el usuario debe proveer un Nick y un password, los cuales el sistema comparara, permitirá o denegara el acceso al mismo, gracias a los cuales el beneficiario podrá realizar sus respectivas búsquedas.

4. USUARIOS.

4.1. USUARIO VISITANTE.

El usuario visitante es un usuario que ingresa al usuario sin autenticación el cual podrá realizar ciertas búsquedas sin obtener los mismos beneficios del usuario registrado, además podrá realizar sugerencias y comentarios sobre el funcionamiento del sistema.

Menú usuario visitante.

HOME
BUSQUEDA GENERAL
SUGERENCIAS Y COMENTARIOS
REGISTRATE



Figura 62. Menú de usuario visitante

4.2. USUARIO REGISTRADO

El usuario registrado al compararse con el usuario visitante tiene la ventaja que puede hacer búsquedas más detalladas y también hacer que el sistema calcule las coordenadas precisas de la porción del plano de Quito, y mostrar un resultado final comparando el croquis de la ciudad con las fotografías satelitales.

El usuario registrado al igual que el usuario visitante puede realizar sugerencias y comentarios sobre el funcionamiento del sistema.

4.3. USUARIO ADMINISTRADOR.

El usuario administrador es el encargado de poder hacer que todo el sistema funcione este usuario ingresa la información, relaciona, actualiza y elimina los datos, además este es capaz de revisar las consultas de todos los resultados finales para que el usuario final tenga los datos más apropiados a su búsqueda.

Este usuario puede administrar toda la información referente al sistema como las calles, lugares y barrios, como también tiene acceso a los datos de los usuarios del sistema pudiendo darles mejores privilegios a ellos como eliminarlos como beneficiarios del sistema.

Menú Usuario administrador

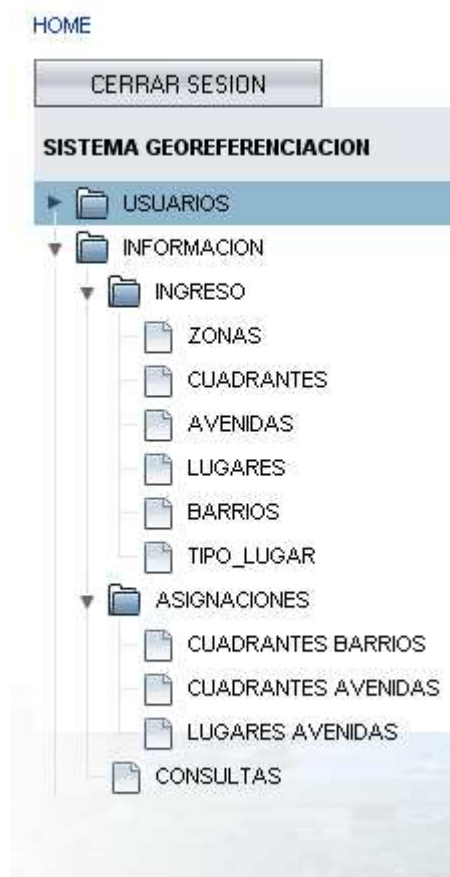


Figura 63. Menú usuario administrador

4.4. REGISTRO DE USUARIOS.

Cada individuo que necesite mejores características de la ciudad de Quito tiene que seguir la opción de registro. La opción de registro es para aquellos usuarios que no tienen cuenta de ingreso al sistema.

Esta opción se encuentra en la interfaz principal en la cual existe un enlace de registro. En la cual las personas ingresarán sus datos que solo servirán para saber qué tipo de usuario ingresa en nuestro sistema (esta información es básica sin invadir la privacidad de cada usuario).

Los datos que el sistema pide para el registro son los siguientes:

- Nombres.
- Usuario
- Password
- E-mail
- País
- Ciudad

Siendo usuario el nombre el Nick con el que podrá ingresar a la cuenta.

Figura 64. Registro de nuevos usuarios

4.5. ADMINISTRACIÓN DE CONTRASEÑAS

Cada usuario del sistema debe tener una cuenta y una contraseña con la cual se debe autenticar para posterior uso del sistema esta clave puede ser administrada únicamente por el usuario que la posee. El sistema le da la capacidad de cambiarla pidiendo como parámetro solo la clave anterior y una nueva y posteriormente la confirmación de la nueva contraseña.

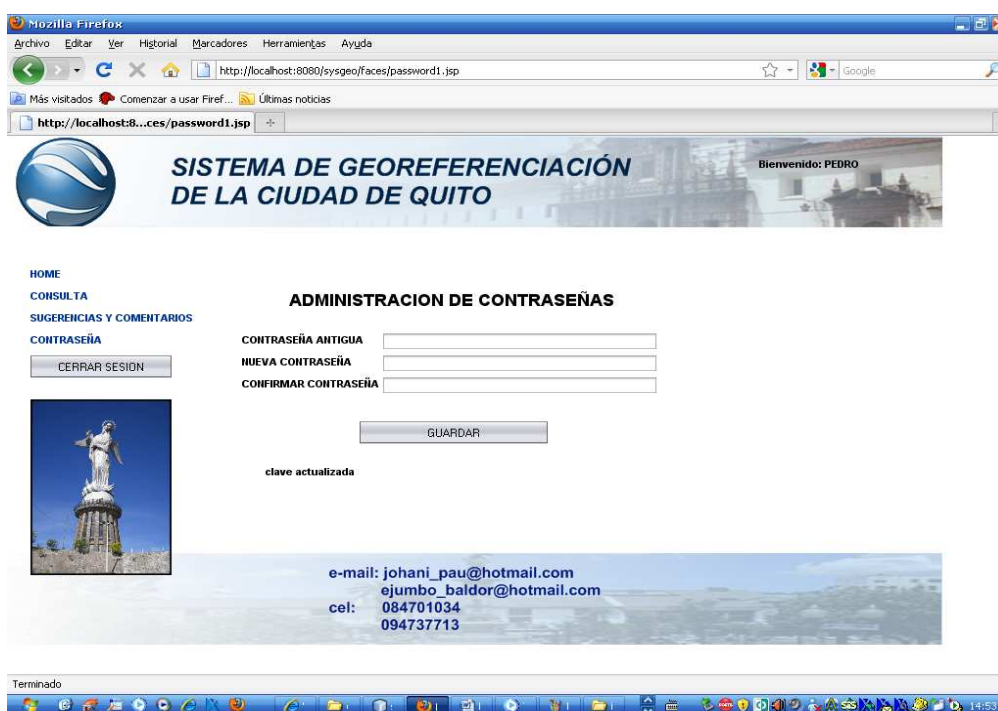


Figura 65. Administración de contraseñas

5. ENLACES

5.1. HOME

El enlace home permitirá que los usuarios puedan dirigirse a la página principal del sistema en la cual explica que actividad realiza el sistema que puede encontrar el usuario, las ventajas y desventajas del mismo.

La página principal dependiendo del tipo de usuario mostrara enlaces a las diferentes tareas que puede realizar el sistema.

5.2. BÚSQUEDAS GENERALES

Búsquedas generales es una actividad que solo puede realizar el usuario visitante la búsqueda es de una manera general sin la vista de la imagen satelital que provee el sistema Google Earth.

Para realizar este tipo de búsqueda el usuario debe ingresar a través del enlace búsqueda general, en el cual se debe colocar un criterio de búsqueda calle, barrio o lugar que desea localizar.

Dependiendo del criterio de búsqueda los resultados proporcionados por el sistema serán las calles, los lugares o barrios que más se acerquen a los datos proporcionados por el usuario. Dando el nombre de lo que se desea localizar y el cuadrante del plano de la ciudad donde se encuentra. Con lo cual el usuario tendrá que decidir a qué cuadrante desea acceder mediante un clic de enlace que aparecerá junto con el resultado del sistema.

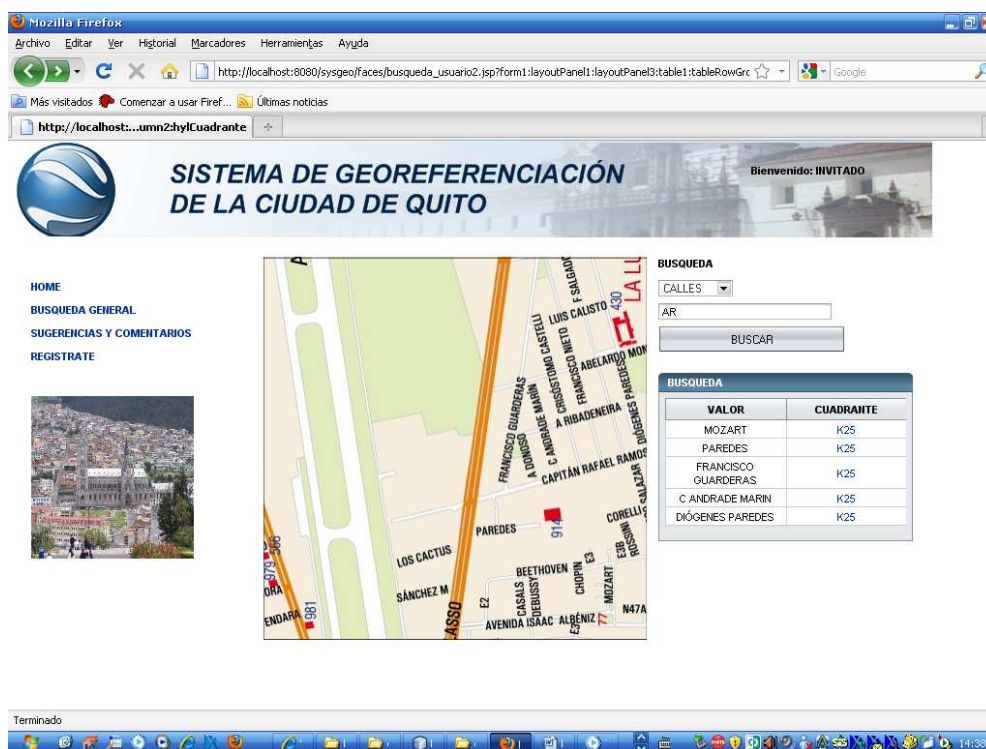


Figura 66. Búsquedas generales

Con los permisos para este tipo de usuario posee solo se permita ver esta única forma de resultado por parte del sistema.

5.3. BÚSQUEDAS CON PRIVILEGIOS

La búsqueda con privilegios solo la pueden realizar las personas que se autenticuen en el sistema, tanto los usuarios registrados como administradores tendrán en sus interfaces un enlace llamado **consultas** el cual les guiara a una interfaz con tres diferentes criterios de búsqueda que pueden ser por calles, por lugares o por barrios, con el valor que el usuario desee encontrar en el sistema.

La respuesta al igual que la de búsqueda sin privilegios será el de una cuadrante del plano de Quito el cual el usuario podrá elegir cuál de ellos observar.

Esta pantalla tiene la ventaja de poder calcular las coordenadas del plano y ubicarlo sobre la interfaz del Google earth, solo al presionar el botón con la etiqueta “GOOGLE EARTH” que se encuentra junto con la respuesta del sistema con lo cual el usuario podrá tener mejores guías referentes a su búsqueda.



BUSQUEDA

CALLES ▾

A

BUSCAR

BUSQUEDA

| VALOR | CUADRANTE |
|-----------------------|-----------|
| LOS CACTUS | K25 |
| SÁNCHEZ M | K25 |
| AVENIDA ISAAC ALBÉNIZ | K25 |
| CASALS | K25 |
| MOZART | K25 |

Página: 1 de 4

GOOGLE EARTH

Figura 67. Búsquedas con privilegios

Para que se pueda activar esta acción y observar los resultados entre las imágenes satelitales y el plano de la ciudad de Quito es necesario tener instalado el sistema Google earth en el computador.

Una vez que se activado esta acción se ejecutara el programa google earth mostrando dirigiéndose hacia la coordenada que calculo el sistema.



Figura 68. Llamado al sistema google earth

Gracias a las coordenadas calculadas y a través de las imágenes del sistema del plano de la ciudad de Quito se puede obtener una resultado entre las imágenes satelitales y el plano de Quito mostrando un resultado interactivo para el usuario.

La cual permite trabajar con la transparencia de la imagen cuando esta herramienta esta en 100% mostrara una imagen solida pero cuando este porcentaje baja la transparencia empieza a mostrarse dando una concatenación entre las imágenes. Satelitales y la de los planos de la ciudad.

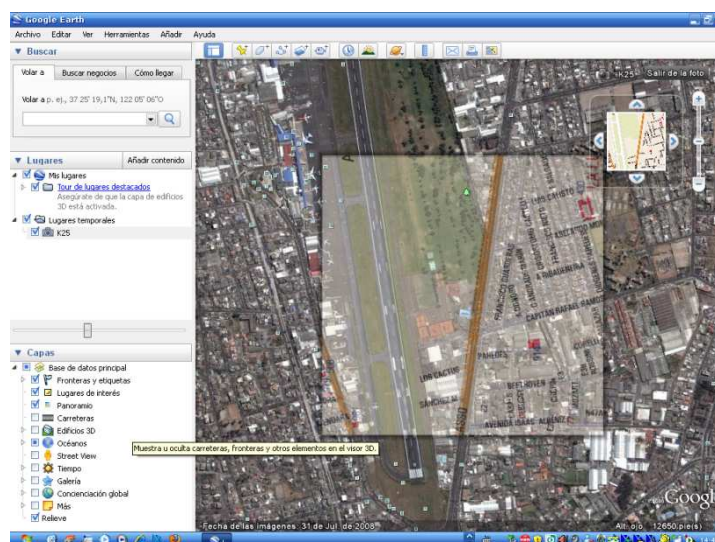


Figura 71. Slider de transparencias

5.4. ADMINISTRACIÓN DE USUARIOS

La administración de la información solo puede ser realizada por el usuario administrador.

Entre la información ingresada están los datos de los usuarios pudiendo el administrador manipular todos aquellos estén registrado a excepción de las claves de ingreso al sistema, inclusive a través de este usuario se puede dar mayores privilegios (convertirlo en otro usuario administrador).

El usuario administrador puede dar más privilegios también es capaz de quitarlos o de eliminarlos como beneficiarios del sistema.

El administrador tendrá acceso a los usuarios a través de criterios de búsqueda pudiendo realizar búsquedas por ciudad, o país al que pertenece.

Esta opción se encuentra en el menú usuarios enlace búsquedas. Mientras que si el administrador desea crear nuevos usuarios puede hacerlo mediante el mismo menú en la opción ingreso.

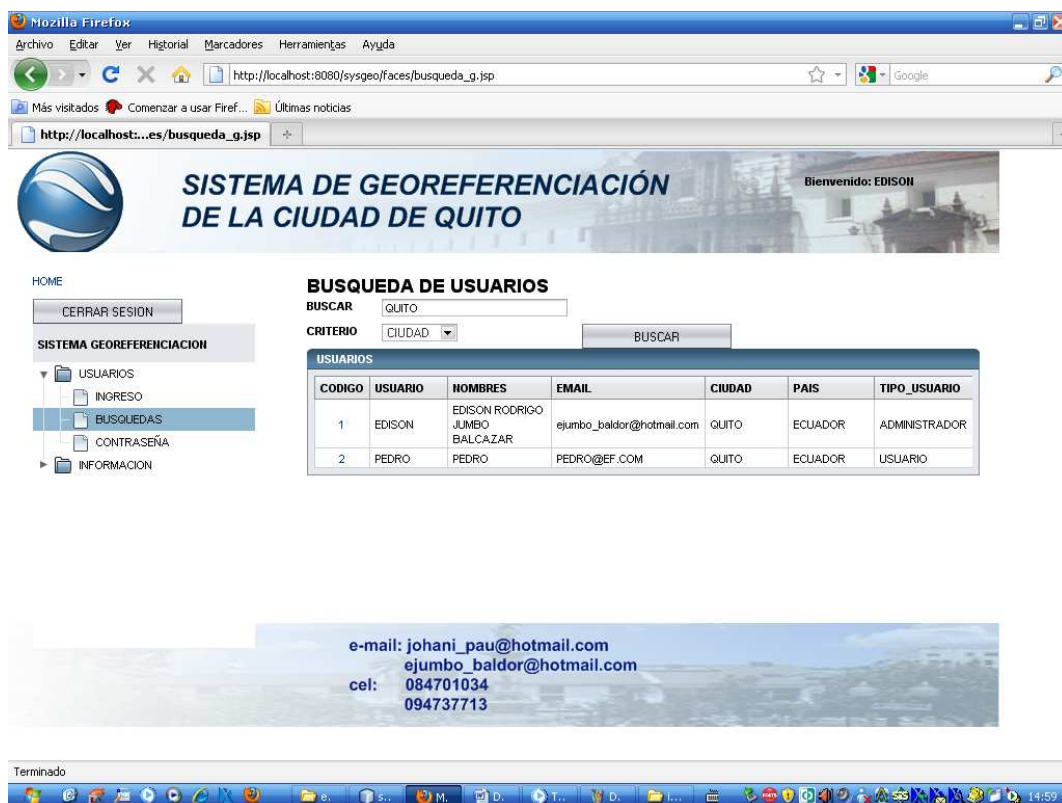


Figura 72. Administración de usuarios

Todos los resultados de las consultas tiene un enlace en la tabla que muestran la información y realizando un pequeño clic en este enlace se mostrara los datos del usuario de manera individual la cual puede ser editada y gestionada e inclusive eliminada.

5.5. ADMINISTRACIÓN DE INFORMACIÓN DEL SISTEMA

La información del sistema está organizada por calles, barrios, lugares, tipos de lugares, y cuadrantes.

Estos datos deben ser ingresados por separado, gestionados para luego ser relacionados entre ellos junto con los cuadrantes en donde se encuentran.



Figura 73. Administración de información del sistema

Para ingresar uno de los datos, se debe escoger una de las opciones o enlaces que se encuentran en el menú información submenú ingreso. Cada dato que se ingresa se mostrara en una tabla de resultados la cual si se encuentran errores pueden ser corregidos o eliminados (Cada tabla muestra un enlace con el cual se puede administrar la información).

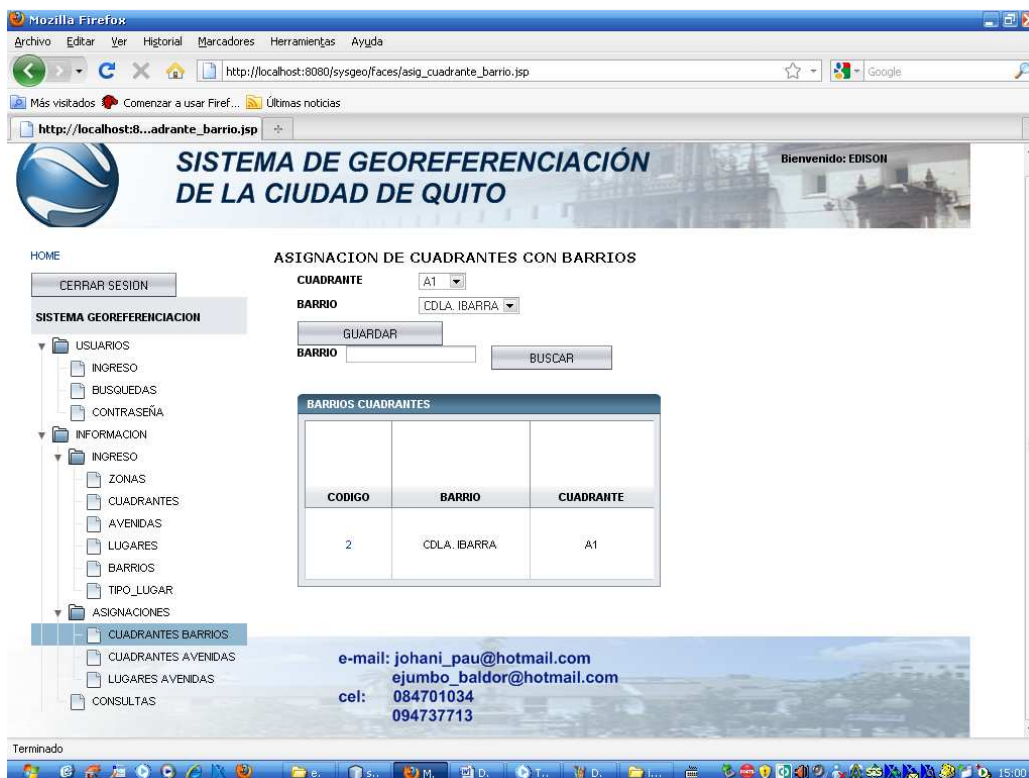


Figura 74. Relación de información

Una vez terminado el ingreso de datos se procede a hacer las relaciones entre la información de la ciudad y el cuadrante en la que se encuentra. Esto se realiza en la interface que están en los enlaces del usuario administrador en la sección de asignaciones. En la que se deben relacionar los barrios, las calles y los lugares con los cuadrantes respectivos, estas asignaciones al igual que la información primaria también puede ser administrada como por ejemplo actualizar un lugar o eliminarlo por defectos de digitación.

Una vez ingresada la información se procede a realizar una relación entre estos datos. Se puede realizar esta tarea en el menú información, submenú asignaciones. En la cual se debe escoger una de las opciones y proceder a la realización por ejemplo la calle con el cuadrante donde se encuentra o el barrio con el cuadrante donde está.



Figura 75. Relación de avenidas con los cuadrantes

5.6. SUGERENCIAS Y COMENTARIOS.

Las sugerencias y comentarios no solo las puede colocar un usuario visitante si no también los usuarios registrados. Estas indicaciones proporcionadas servirán para futuras modificaciones del sistema, siendo una manera de retroalimentación para poder dar un mejor servicio a la colectividad.

SUGERENCIAS Y COMENTARIOS

Ingrese su comentario:

Figura 76. Sugerencias y comentarios

MANUAL DE INSTALACIÓN DE GLASSFISH

INSTALACIÓN DE GLASSFISH

Glassfish es el servidor por defecto de la Sun Microsystems para las aplicaciones web JSF. Las configuraciones que se deben realizar para el funcionamiento de este servidor son mínimas pero se deben de seguir en su totalidad para obtener el resultado deseado.

- Instalar la maquina virtual de java. Tener en cuenta que versión se debe instalar para que JSF pueda funcionar correctamente.
- Crear las variables de entorno de java, uno de los pasos más simples de realizar esta tarea es en las propiedades del sistema pestaña opciones avanzadas/variables de entorno.

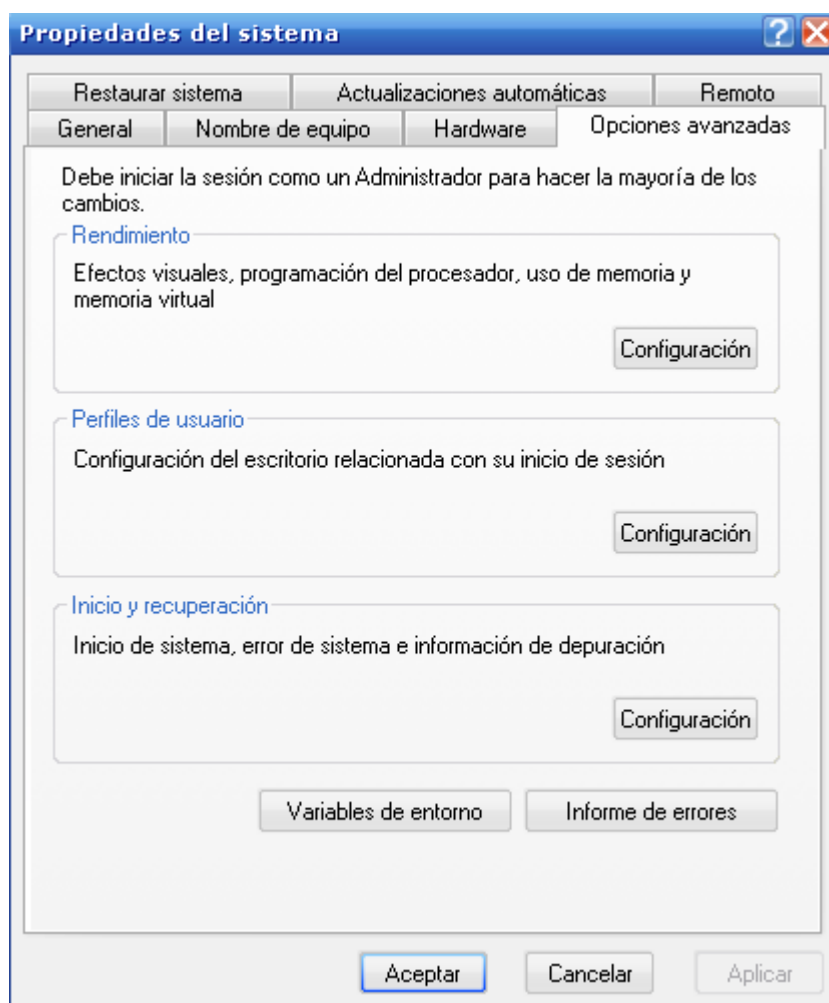


Figura 77. Propiedades del Sistema

- La primera variable de entorno debe de tener como nombre JAVA_HOME y en el valor de la variable se debe colocar el Path en donde se instalo Java por ejemplo “c:\Archivos de programa\java\jdk1.6.0_01\”. La segunda variable de entorno con la que se debe contar es JDK y el valor de esta variable debe de ser la misma que la primera variable.

Para que no exista inconvenientes con las variables de entorno se recomienda crearlas como variables de usuario y variables del sistema.

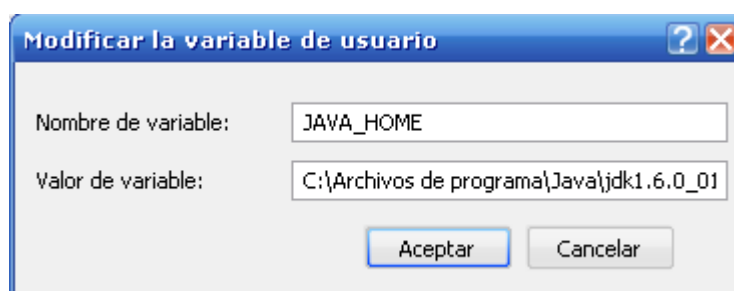


Figura 78. Variables de entorno

- El servidor GlassFish se puede descargar del sitio de la Sun Microsystems o directamente de la página del mismo proyecto. (<https://glassfish.dev.java.net/>)
- El fichero que se utilizo para el funcionamiento de las interfaces JSF es el (glassfish-installer-v2.1-b50-windows.jar) el cual se utiliza para windows.
- Cuando se descargen este fichero se debe de copiar en la unidad “C:\” Con lo se debe de abrir el símbolo del sistema y colocar la siguiente línea C:\ java -Xmx256 -jar glassfish-installer-v2.1-b50-windows-jar. O a su vez realizar doble clic sobre el archivo con lo deberá de esperar unos segundos para que se muestre la interfaz la que contiene la licencia del servidor. El botón de **ACCEPT** aparece desactivado para que se active se debe de llevar la barra de desplazamiento vertical hasta el final.

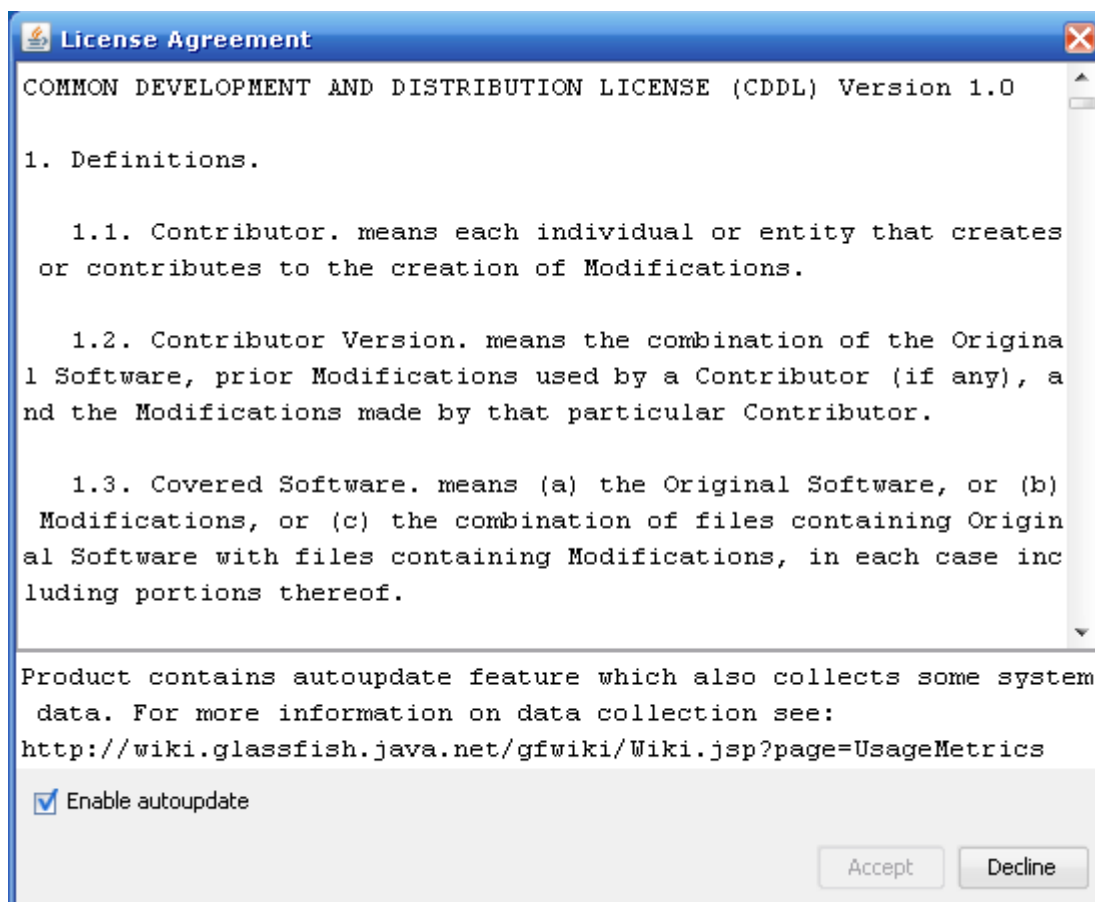


Figura 79. Licencia del servidor

- Se creara automáticamente un directorio “**C:\glassfish**” en el cual se instala el servidor.
- Para que el servidor empiece a funcionar se debe de crear un dominio. Compilando el archivo **setup.xml** que se encuentra en el directorio. Para realizar este compilado se debe de tener el compilar ANT instalado en el

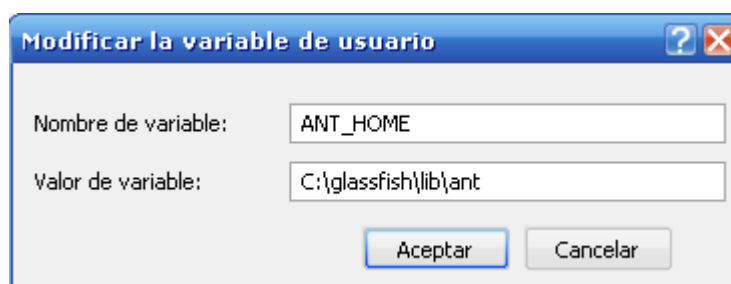


Figura 80. Variable de entorno ANT

computador, si no se posee este compilador GLASSFISH lo incorpora solo teniendo que crear una variable de entorno tanto en las variables de usuario como las del sistema, en el valor de la variable colocar el Path (C:\glassfish\lib\ant)

- Antes de realizar la compilación se debe tener en cuenta que el archivo setup.xml trae el nombre del dominio, el nombre de usuario, password, y los puertos en los que va a funcionar. Por defecto aparecen los siguientes.

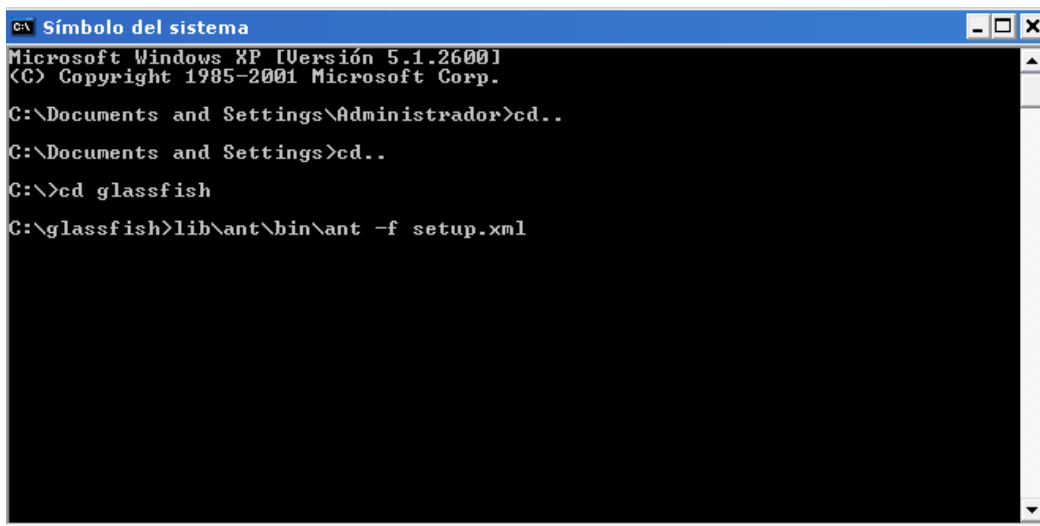
```
<property name="domain.name" value="domain1"/>
<property name="instance.name" value="server"/>
<property name="admin.user" value="admin"/>
<property name="admin.password" value="adminadmin"/>
<property name="master.password" value="changeit"/>
<property name="admin.port" value="4848"/>
<property name="instance.port" value="8080"/>
<property name="orb.port" value="3700"/>
<property name="imq.port" value="7676"/>
<property name="https.port" value="8181"/>
```

- Estos pueden ser cambiados por el usuario como se realizo en esta oportunidad lo único que se dejo por defecto fue el nombre del usuario (admin) y el password (adminadmin) mientras que los puerto donde funcionan fueron cambiados a lo siguiente:

```
<property name="admin.port" value="5858"/>
<property name="instance.port" value="8083"/>
<property name="orb.port" value="3701"/>
<property name="imq.port" value="7677"/>
<property name="https.port" value="8183"/>
```

- Si los puertos fueron cambiados posteriormente tendrán que ser utilizados cuando se levante una aplicación en el servidor.

- Una vez que se ha modificado el archivo setup.xml se procede a la compilación, colocando en el símbolo del sistema en el directorio GLASSFISH la siguiente línea. lib/ant/bin/ant -f setup.xml.



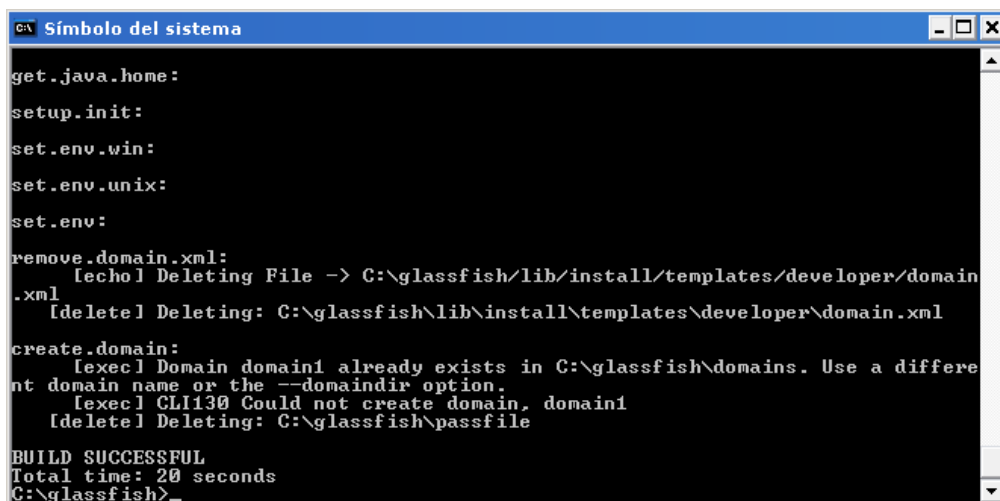
```

C:\> Símbolo del sistema
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrador>cd..
C:\Documents and Settings>cd..
C:\>cd glassfish
C:\glassfish>lib\ant\bin\ant -f setup.xml
  
```

Figura 81. Compilación del archivo setup.xml con ANT

- Cuando la compilación se ha realizado de una manera exitosa se crea un directorio con el nombre del dominio en este caso **domain1**.
- Para levantar el servicio se debe de moverse hasta el directorio "c:/glassfish/bin" y utilizar las siguiente linea de codigo.
asadmin start-domain domain1.



```

C:\> Símbolo del sistema

get.java.home:
setup.init:
set.env.win:
set.env.unix:
set.env:
remove.domain.xml:
    [echo] Deleting File -> C:\glassfish\lib\install\templates\developer\domain.xml
    [delete] Deleting: C:\glassfish\lib\install\templates\developer\domain.xml
create.domain:
    [exec] Domain domain1 already exists in C:\glassfish\domains. Use a different domain name or the --domaindir option.
    [exec] CLI130 Could not create domain, domain1
    [delete] Deleting: C:\glassfish\passfile

BUILD SUCCESSFUL
Total time: 20 seconds
C:\glassfish>
  
```

Figura 82. Creación del dominio

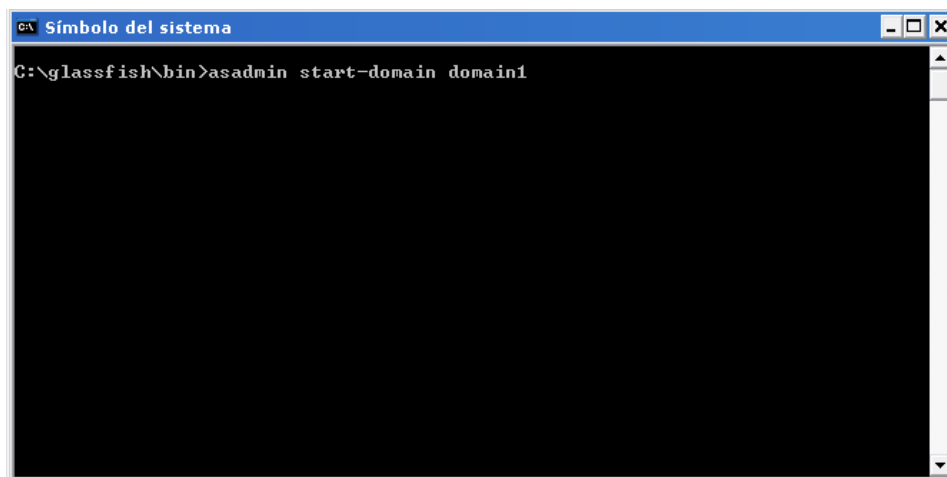


Figura 83. Levantar el Servicio

- Para ingresar a la configuración del sistema se debe abrir el navegador que tenga instalado en su equipo y colocar la siguiente dirección “http://localhost:5858/ “ no olvidarse que el puerto es el que se cambio en los pasos anteriores.

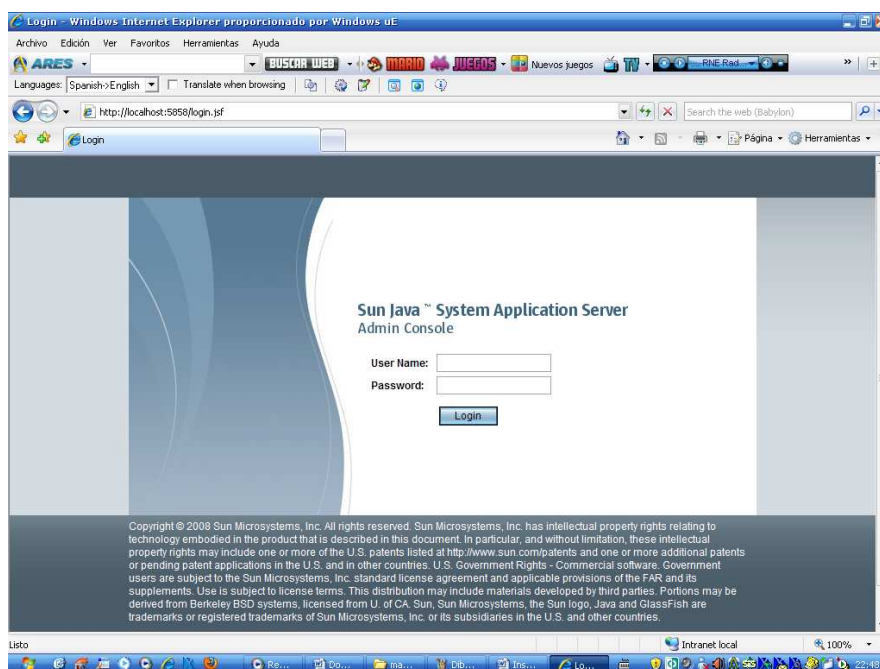


Figura 84. Interfaz de autenticación de la configuración de Glassfish

- La interfaz principal en la se configura glassfish es la siguiente:

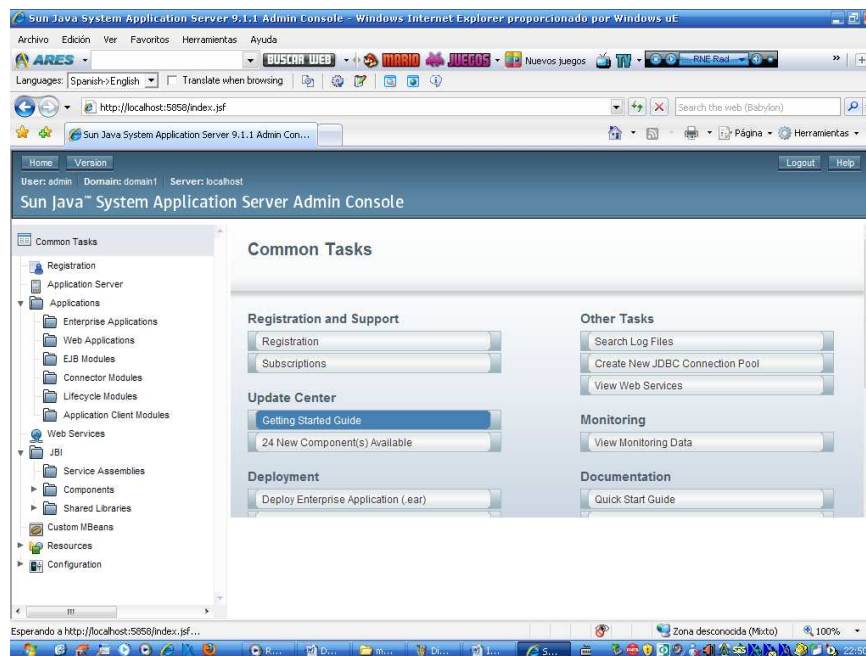


Figura 85. Interfaz principal de glassfish

- Una de las configuraciones que se pueden realizar es cargar las diferentes aplicaciones entre las que tenemos **web Applications** donde se pueden administrar todas las aplicaciones web.

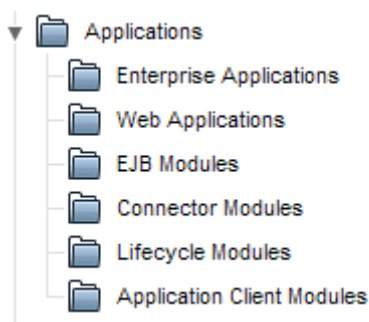


Figura 86. Aplicaciones

- Eligiendo esta opción se puede cargar un archivo war que son los archivos que tienen todas las configuraciones y archivos de las aplicaciones web de java.

- Para cargar este archivo se realiza clic en **Deploy** en el cual se debe localizar donde se encuentra es archivo.

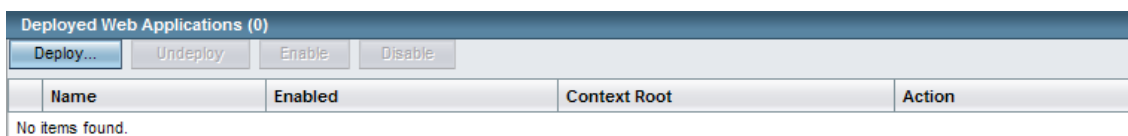


Figura 87. Administrador de aplicaciones web

- La ubicación del archivo war no necesariamente debe de estar en la carpeta de inicio del servidor, esto solo se realiza cuando no se cuenta con el fichero war.

Applications > Web Applications

Deploy Enterprise Applications/Modules

Specify the location of an application to deploy. Applications can be in packaged files such as .war, .ear, .jar, and .rar.

Type: Web Application (.war)

Location: ☒ Packaged file to be uploaded to the server

C:\Ejemplo.war

☐ Local packaged file or directory that is accessible from the Application Server

General

Application Name: * Ejemplo

Context Root: Ejemplo
Path relative to server's base URL

Virtual Servers:

Associates an internet domain name with a physical server

Figura 88. Interfaz de carga de aplicaciones web

- Cuando una aplicación war a sido cargada de manera correcta aparece el nombre y si esta activado o no, a partir de ese momento la aplicación puede ser administrada desde esta interfaz.

| Deployed Web Applications (1) | | | | |
|---|---------|---------|--------------|---|
| <input type="checkbox"/> <input type="checkbox"/> Deploy... Undeploy Enable Disable | | | | |
| | Name | Enabled | Context Root | Action |
| <input type="checkbox"/> | Ejemplo | true | Ejemplo | Launch Redeploy |

Figura 89. Aplicación web cargada

- Para ver el funcionamiento de nuestro ejemplo se coloca en el navegador. **“http://localhost:8083/ejemplo”** con lo que se comprueba el funcionamiento de nuestro servidor, listo para aceptar cualquier aplicación web JSF.

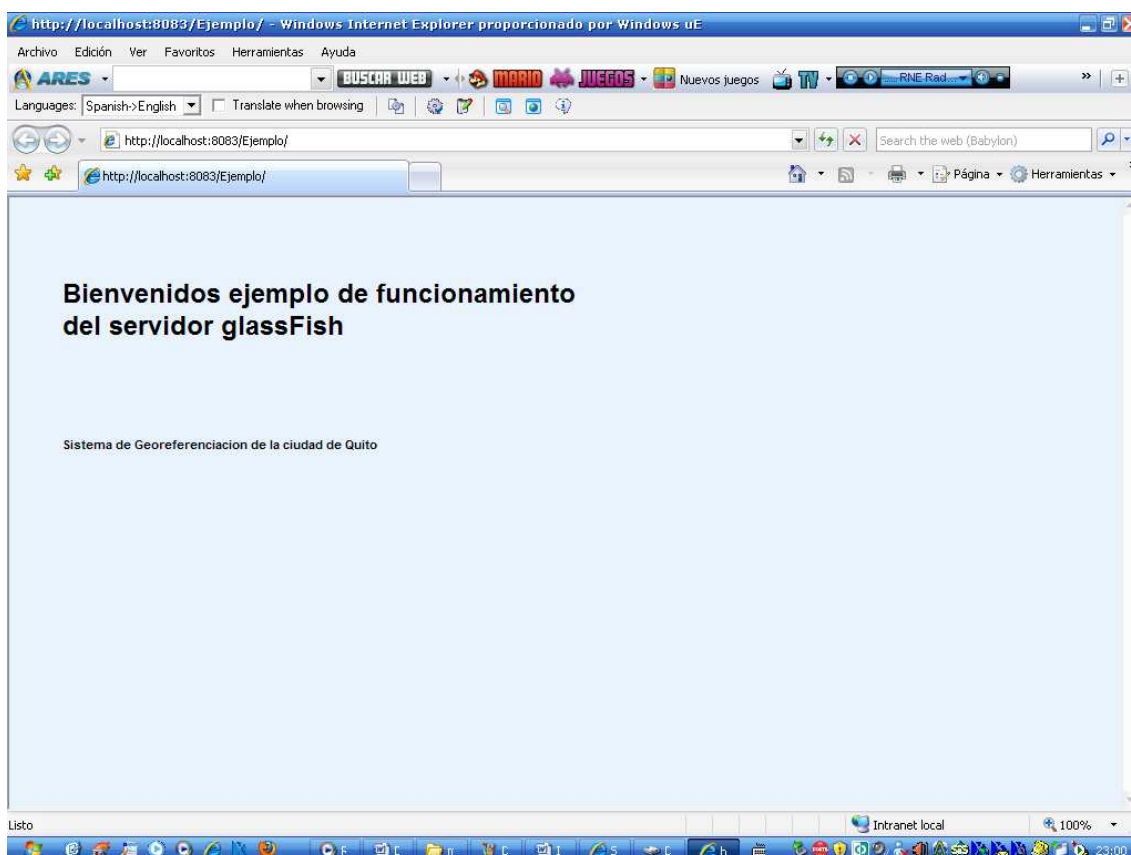


Figura 90. Ejemplo funcionamiento JSF